ISO TC184/SC4/WG12 N463

Date:1999-12-14

Supersedes ISO TC 184/SC4/WG12 N279

ISO/DIS 10303-44 Second Edition
Product data representation and exchange: Integrated generic resource: Product structure configuration

ABSTRACT:

This Part of ISO 10303 specifies the integrated generic resources used for representing the information needed to define the structure of a product and to manage the configuration of that product. It defines data structures to maintain information for bill-of-material, parts list, and manufacturing planning applications.

KEYWORDS:

integrated generic resource, product structure, configuration management, effectivity, product concept, make from usage, assembly component usage

COMMENTS TO READER:

This document is the editors draft version of the IS stage of ISO 10303-44.

This is the proposed second edition of ISO 10303-44. The content of this document incorporates the resolutions approved by WG12 to SEDS issues raised against the 1994 edition and NWI/CD ballot comments raised against the WG12 N102 document. These changes are intended to ensure strict "upward compatibility" with the previous edition.

Details of the differences between this document and the 1994 edition are given in WG12 N098 and the CD ballot comment issues log.

Project Leader: Mitch Gilbert
Address: P.D.I.T., Inc.
4130 Faber Place Drive, Suite 303
Charleston, SC 29405 USA
Telephone: +1-843-747-1420
Telefacsimile: +1-843-747-2438
Electronic mail: mgilbert@pdit.com

Project Editor: Diane Craig
Address: P.D.I.T., Inc.
100 W. Broadway, Suite 540
Long Beach, CA 90802 USA
Telephone: +1-562-495-6500
Telefacsimile: +1-562-495-6509
Electronic mail: dcraig@pdit.com

International Organization for Standardization
 Case Postale 56 • CH-2111 Geneve 20 • Switzerland

Contents

Page

Figures

Tables

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75% of the member bodies casting a vote.

International Standard ISO 10303-44 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC4, *Industrial data*. This second edition of ISO 10303-44 constitutes a technical revision of the first edition (ISO 10303-44:1994), which is provisionally retained to support continued use and maintenance of implementations based on the first edition, and to satisfy the normative references of other parts of ISO 10303. This edition incorporates the corrections published in ISO 10303-44/Cor.1:1999.

This International Standard is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application interpreted constructs, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1.

A complete list of parts of ISO 10303 is available from Internet:

<http://www.nist.gov/sc4/editing/step/titles/>.

This part of ISO 10303 is a member of the integrated resources series. The integrated resources specify a single conceptual product data model.

Annexes A and B form an integral part of this part of ISO 10303. Annexes C and D are for information only.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

Major subdivisions of this part of ISO 10303 are:

— the **product_structure_schema**;

— the **product_concept_schema**;

— the **configuration_management_schema**.

The **product_structure_schema**:

—defines a product in terms of its composition as a set of constituents or consumed products. These products are defined and related at a specific life cycle stage or discipline view. A product may be assembled from the constituents or produced by consuming other products, or both;

— defines mechanisms for expressing the composition relationship.

The **product_concept_schema** identifies the product concept as a set of specifications for a product derived from analysis of customer needs for the product. It represents the idea of a product based on customer needs and not as it might be designed or built.

The **configuration_management_schema** identifies those products participating in the configuration of another product that is under the direct control of an organization.

The relationships of the schemas in this part of ISO 10303 to other schemas that define the integrated resources of this International Standard are illustrated in Figure 1 using the EXPRESS-G notation. EXPRESS-G is defined in annex D of ISO 10303-11. The **application_context_schema**, **effectivity_-schema**, and **product_definition_schema** are specified in ISO 10303-41. The schemas illustrated in Figure 1 are components of of the integrated resources of this International Standard.

Industrial automation is concerned with the management of information including the following:

— product structure;

— product configuration;

— product change.

Product structure is focused on the aspect of product design that defines a product in terms of a nested decomposition of its constituents. The product structure schema of this part of ISO 10303 and the product

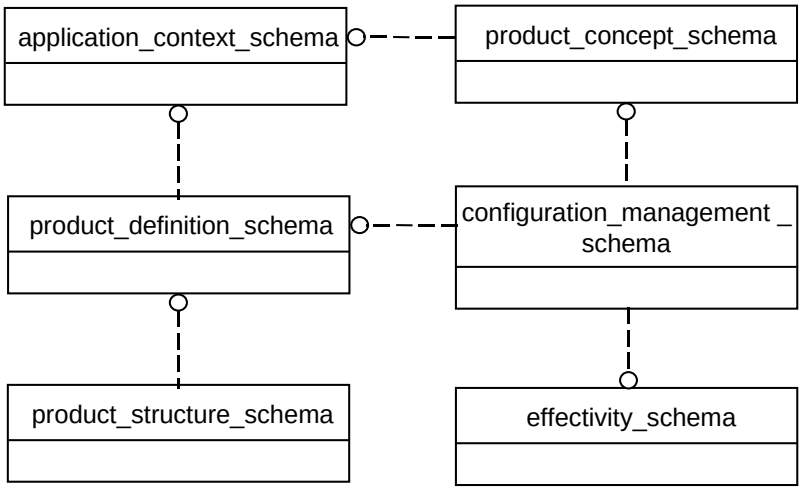

Figure 1 - Relationship of schemas within the integrated resources

definition schema of ISO 10303-41 together define the representation of the information that manages the details of product structure. Examples of the use of the information represented are the generation of bill-of-material reports or the representation of variational bill-of-material structures with conditional part usages.

Product configuration is concerned with the description of the composition of specific products. The planning includes specification of the actual constituents of a product that are to be included in a planned or actual unit. The configuration management schema and the product structure schema represent the information that manages the configuration of a product. The concept of effectivity is used to manage the configuration of a product.

Change management is involved with the changes over time in a product as new versions of a product are developed. This part of ISO 10303 is concerned with changes that affect the organization of constituents into interrelated structures. The configuration management schema represents information on the structural form of the definition of a product as the product changes and is enhanced during the product life cycle. Representation of information on other aspects of change management is defined in the product definition schema and the action schema of ISO 10303-41.

In this International Standard the same English language words may be used to refer to an object in the real world or concept, and as the name of an EXPRESS data type that represents this object or concept. The following typographical convention is used to distinguish between these. If a word or phrase occurs in the same typeface as narrative text, the referent is the object or concept. If the word or phrase occurs in a bold typeface, the referent is the EXPRESS data type.

The name of an EXPRESS data type may be used to refer to the data type itself, or to an instance of the data type. The distinction between these uses is normally clear from the context. If there is a likelihood of ambiguity, either the phrase "entity data type" or "instance(s) of" is included in the text.

This edition of this part of ISO 10303 incorporates modifications that are upwardly compatible with the previous edition. Modifications to EXPRESS specifications are upwardly compatible if:

— the modifications do not result in changes to instances that are encoded according to ISO 10303-21; such instances conform to both the unmodified and modified EXPRESS specifications;

— the modifications do not result in changes to software that conforms to ISO 10303-22 with respect to access to the data content of data structures;

— the modifications do not invalidate mappings to the previous edition of this part that are specified in the mapping table of an ISO 10303 application protocol.

Technical modifications to ISO 10303-44:1994 are categorized as follows: changes to the EXPRESS declarations, new EXPRESS declarations, and changes to definitions.

The following EXPRESS declarations have been modified:

— **alternate_product_relationship**;

— **assembly_component_usage_substitute**;

— **configuration_design**;

— **configuration_effectivity**;

— **configuration_item**;

— **make_from_usage_option**;

— **product_concept**;

— **product_definition_usage**;

— **quantified_assembly_component_usage**;

— **specified_higher_usage_occurrence**.

The following EXPRESS declarations have been added:

— **assembly_component_usage_substitute_with_ranking**;

— **concept_feature_relationship**;

— **concept_feature_relationship_with_condition**;

— **concept_feature_operator**;

— **conditional_concept_feature**;

— **configurable_item**;

— **configuration_design_item**;

— **configuration_item_relationship**;

— **product_concept_feature**;

— **product_concept_feature_association**;

— **product_concept_relationship;**

— **product_definition_occurrence_relationship**.

The definitions of the following EXPRESS data types have been modified:

— **alternate_product_relationship**;

— **assembly_component_usage**;

— **assembly_component_usage_substitute**;

— **configuration_design**;

— **configuration_effectivity**;

— **configuration_item**;

— **make_from_usage_option**;

— **make_from_usage_option_group**;

— **next_assembly_usage_occurrence**;

— **product_concept**;

— **product_definition_usage**;

— **promissory_usage_occurrence**;

— **specified_higher_usage_occurrence**.

Industrial automation systems and integration — Product data representation and exchange — Part 44: Integrated generic resource: Product structure configuration

1 Scope

This part of ISO 10303 specifies the resource constructs to manage the structure and configuration of a product during its life cycle.

The following are within the scope of this part of ISO 10303:

— the relationships among the components and assemblies of products;

— the relationships among products and their components as made by modification of other products;

EXAMPLE 1 The machining of a product from another product is an example of a modification.

— the description of a product as defined by customer needs;

— the dependencies among specifications of a product in order to represent possible product variations to present to a customer;

— the management of the structure for configuration of assemblies and components as planned for manufacture;

— the decomposition of a product to support different product life cycle activities;

EXAMPLE 2 An organization maintains one bill-of-material structure for a product that enumerates the quantity of each component used in each assembly, and into a second bill-of-material that decomposes a product with multiple assemblies into the individual components. See annex E for more examples of different product structure reports that are supported.

— multiple versions of a single product that are equivalent with respect to form, fit, and function.

The following are outside the scope of this part of ISO 10303:

— the relationships among different product definitions for the same product;

NOTE The relationships among different product definitions for the same product are supported by ISO 10303-41.

EXAMPLE 3 The relationship of a product definition for a component in a preliminary design to a corresponding product definition for the same component in a detailed design is an example of this type of relationship.

— administrative activities of the product life cycle including approvals, security classifications, contractual arrangements, and supplier organizations;

— the change process for a product, including the reason for change and what aspect of a product has changed;

— the decisions made, and their reasons, during the product life cycle;

— the physical connections among components of a product;

— the properties that a product constituent may have;

NOTE 1 A mechanism is defined in the **product_property_definition_schema** in ISO 10303-41 to support the association of properties with components. The actual associations are included in various application protocols which are parts of this International Standard. For example, the details of what a material property is and how it is defined are out of scope, as well as the fact that a component has a material property.

— multiple versions of a single product that are not form, fit, and function equivalent.

NOTE 2 The concept of versions of a product is defined in the **product_definition_schema** in ISO 10303-41.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 8824-1:1995, *Information technology — Open systems interconnection — Abstract syntax notation one (ASN.1) — Part 1: Specification of basic notation.*

ISO 10303-1:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles.*

ISO 10303-11:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual.*

ISO/DIS 10303-41:—[1)], *Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resource: Fundamentals of product description and support.*

[1)] To be published.

3 Terms, definitions, and abbreviations

3.1 Terms defined in ISO 10303-1

For the purposes of this part of ISO 10303, the following terms and definitions given in ISO 10303-1 apply.

— assembly;

— component;

— product.

3.2 Terms defined in ISO 10303-41

For the purposes of this part of ISO 10303, the following terms and definitions given in ISO 10303-41 apply.

— agreement of common understanding;

— annotated EXPRESS schema;

— type of product.

3.3 Other definitions

For the purposes of this part of ISO 10303, the following terms and definitions apply:

3.3.1
ancestor node
any node that can be reached from a given node, by successive traversals of links in the reverse direction. For a given node, its ancestor nodes include all parent nodes, all parent nodes of these parent nodes, etc.

3.3.2
bill-of-material (BOM) data structure
a graph of data elements that depict a bill-of-material structure.

NOTE See annex E for examples of bill-of-material data structures.

3.3.3
bill-of-material (BOM) structure
a description of the structure of a product in terms of its nested constituents.

NOTE See annex E for examples of bill-of-material structures.

3.3.4
child node
the node to which a link is pointing.

3.3.5
configuration item
a subdivision of a product, either a component or an assembly.

3.3.6
descendent node
any node that can be reached from a given node, by successive traversals of links. For a given node, its descendent nodes include all children nodes, all children nodes of these children nodes, etc.

3.3.7
directed acyclic graph (DAG)
a collection of nodes and directed links such that no node is an ancestor (or descendant) of itself.

3.3.8
effectivity
a characteristic that indicates whether a product is allowed, approved, or permitted to be used in another product.

3.3.9
form, fit, and function
a collection of characteristics that include the shape of a product, the way it interfaces with other products, and the purpose that the product serves.

3.3.10
leaf node
a node that has no children.

3.3.11
link
a uni-directional relationship from one node to another node within a directed acyclic graph.

3.3.12
lot
batch
a collection of distinct products that are treated as a single unit.

EXAMPLE Three thousand bundles of yarn are divided into different groups. Each group is submerged in a separate batch of red dye. The group is treated as a lot and assigned a lot number. The lot number is identified so that conditions causing slight changes in the colour are differentiable among bundles belonging to different lots. A customer may wish to purchase bundles of the same lot to ensure consistency of the colour.

3.3.13
node
an element of a directed acyclic graph, connected to other such elements by links.

3.3.14
parent node
the node from which a link is initiated.

3.3.15
parts list data structure
a graph of instances that depict a parts list structure.

NOTE See annex E for examples of parts list data structures.

3.3.16
parts list structure
a description of the structure of a product in terms of a hierarchy of all distinct usages of its constituents.

NOTE See annex E for cases of parts list structures.

3.3.17
promissory use
the intention to use a constituent in an assembly.

3.3.18
root node
a node that has no parents.

3.3.19
tree
a restricted type of directed acyclic graph in which there is only one root node, and in which each node has at most one parent.

3.4 Abbreviations

For the purposes of this part of ISO 10303, the following abbreviations apply.

BOM bill-of-material

CMconfiguration management

4 Product structure

The following EXPRESS declaration begins the **product_structure_schema** and identifies the necessary external references.

EXPRESS specification:

```
*)
SCHEMA product_structure_schema;
```

```
  REFERENCE FROM product_definition_schema
    (product,
     product_definition,
     product_definition_relationship,
     acyclic_product_definition_relationship);

  REFERENCE FROM measure_schema
     (measure_with_unit);

  REFERENCE FROM support_resource_schema
     (identifier, label, text);
(*
```

NOTE 1The schemas referenced above are specified in the following part of ISO 10303:

product_definition_schemaISO 10303-41

measure_schemaISO 10303-41

support_resource_schemaISO 10303-41

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

4.1 Introduction

The subject of the **product_structure_schema** is the relationship between the definitions of:

— products that are assembled to make other products;

— products that are changed to make other products.

These relationships are defined as specializations of a general relationship among **product_definition**s as specified in the **product_definition_schema** in ISO 10303-41. In addition, any product that is used in a product structure may have alternative products specified for that use. Collectively these relationships are referred to as a product structure.

The product structure defines the different methods by which a product can be represented as being made up of constituents. Product structure relationships are established among the constituents that make up a product. Product structure is an aspect of product definition.

The relationships between product definitions are represented in this part of ISO 10303 using a subtype of the **product_definition_relationship** entity as defined in the **product_definition_schema** in ISO 10303-41. The subtypes of **product_definition_relationship** defined in this schema establish additional constraints and meanings to the supertype in ISO 10303-41.

NOTE Figure 2 shows a partial view of the **product_definition_schema** in ISO 10303-41 and depicts the subtype structure of entities defined in this schema. Attributes of the entities defined in this schema and of the **product_-definition** entity are not shown. See Figure D.1 for a complete diagram showing all attributes and entities of this schema.

The product structure concept applies to multiple definitions of the structure of a single product version. The different definitions correspond to different organization requirements for defining the structure of a product during the life cycle of the product's development.

EXAMPLE An organization may define a bill-of-material structure for both a design engineering release life cycle activity, as well as for a manufacturing engineering activity.

This schema supports the concept of making a product from another product. This concept deals with the relationship between a product and the results of a process applied to that product which produces a new product.

4.2 Fundamental concepts and assumptions

The following concepts and assumptions apply.

— Product structures are modelled by directed acyclic graphs (DAG). In these models, nodes represent product definitions, and the directed links represent composed-of relationships. In this schema, nodes correspond to **product_definition** entities and the links correspond to **assembly_component_usage** entities.

NOTE 1For a detailed discussion of graph theory, see citation [1] in the bibliography.

— Many forms of product structure can be represented using this schema. Two product structures of special utility are bill-of-material and parts list structures.

NOTE 2 Examples and diagrams of the manner in which the entities of this part of ISO 10303 may be used to represent product structures are included in annex E.

— A parts list structure is a specific form of a bill-of-material that can be represented by a tree. A bill-of-material structure may require a more general DAG.

— For a general product structure, in order to identify the usage of any constituent within an assembled product, it is necessary to identify the path between the assembled product and the constituent. The **specified_higher_usage_occurrence** entity provides this capability.

4.3 Product structure entity definitions

4.3.1 alternate_product_relationship

The alternate product relationship is an association between two products whereby one product, the alternate product, may be used in place of another product, the base product.

When one product is an alternate for another product it is understood that there is no interest to keep track of which product, the base or any alternates specified, is used as a particular instance of the base product within a product structure.

NOTE An organization may track design changes for a base part, and establish effectivity conditions for the use of that base part in various assemblies to be manufactured. The use of an alternate product implies that an

organization does not specify any particular version of the alternate product or establish effectivities relating to it.

An **alternate_product_relationship** for which the base product is an assembly specifies that the entire product structure of the base product may be used in place of the alternate product and its product structure.

EXAMPLE 1 Two bolts of the same size are products. One bolt has a square head and the other has a hexagonal head. These head shapes are properties of the respective products. The two bolts are considered equivalent with respect to form, fit, and function: they both have sufficiently close physical shape, they take up the same space when used, and they both serve to fasten two things together. Thus, one of these two bolts could be considered to be an alternate part for the other bolt..

The relationship established by the **alternate_product_relationship** need not be symmetric. If B is an alternate product for A, A is not an implicit alternate product for B.

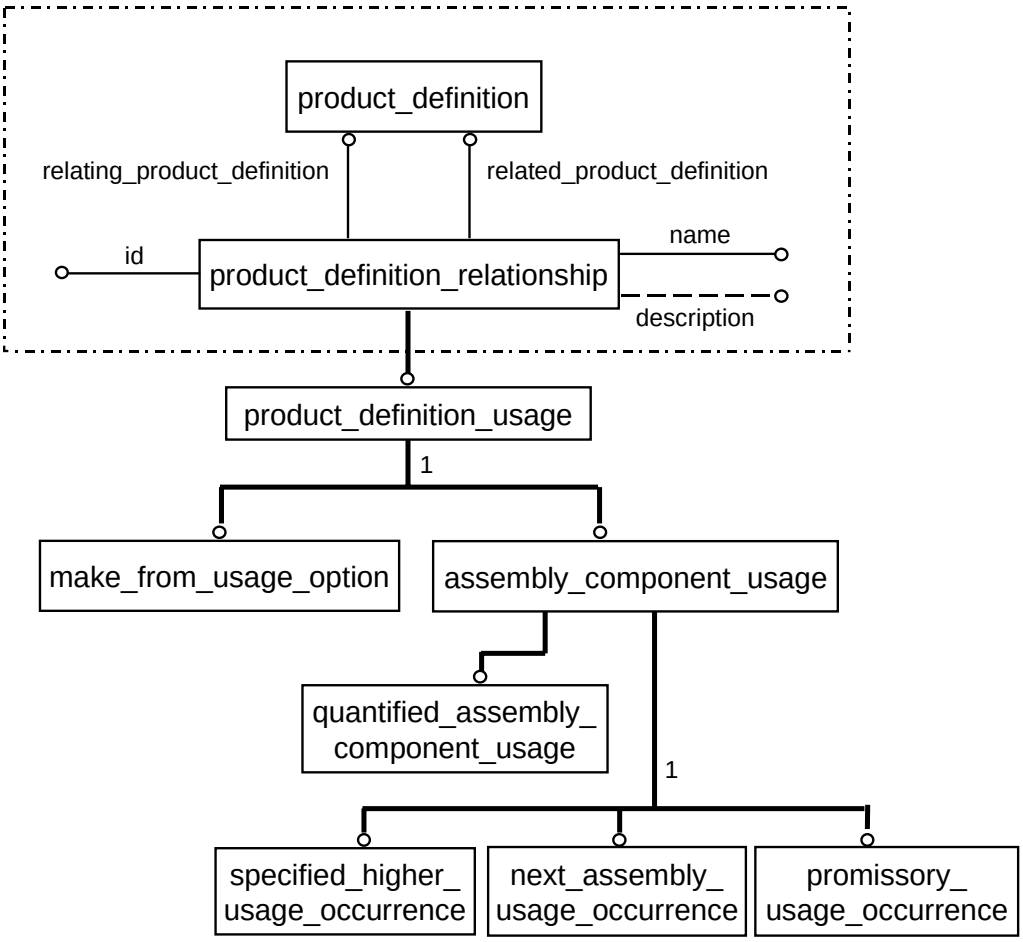

Figure 2 - Relationship of product structure entities to ISO 10303-41

EXPRESS specification:

```
*)
ENTITY alternate_product_relationship;
  name        : label;
  definition  : OPTIONAL text;
  alternate   : product;
  base        : product;
  basis       : text;
UNIQUE
  UR1: alternate, base;
WHERE
  WR1: alternate :<>: base;
END_ENTITY;
(*
```

Attribute definitions:

name: the **label** by which the **alternate_product_relationship** is known.

definition: the **text** that characterizes the **alternate_product_relationship**. The value of the definition need not be specified.

alternate: a product that may be used in place of the base product.

base: a product for which another product may be used as an alternate.

basis: a text description to specify the rationale and criteria used to consider the alternate product in place of the base product.

EXAMPLE 2 In the **alternate_product_relationship** for the two bolts defined in example 1, the value of the basis attribute, could be "head shape", while the value of the definition attribute could be "alternate for use as fastener in engine assembly".

Formal propositions:

UR1: The combination of the alternate product and the base product shall be unique.

WR1: The instance for the alternate product shall not be the same as the instance of the base product.

4.3.2 assembly_component_usage

The assembly component usage relates a constituent to its assembly. The **assembly_component_usage** entity is a subtype of the **product_definition_usage** entity that should be used to establish a relationship between **product_definition**s within one of the following three product structures:

— bill-of-material (BOM) structure;

NOTE 1 In a BOM structure, **product_definition** entities represent nodes and **next_assembly_usage_occurrence** or **quantified_assembly_component_usage** entities represent links.

— parts list structure;

NOTE 2 In a parts list structure, a **product_definition** entity represents the root node. The **next_assembly_-usage_occurrence** entities represent nodes at each intermediate level of the structure. The **specified_higher_-usage_occurrence** entities enable links to higher-levels of the structure.

— promissory use structure.

NOTE 3 In a promissory use structure, **product_definition** entities represent nodes, and **promissory_usage_-occurrence** entities represent links between the nodes.

NOTE 4 In order to allow for extensions to this entity in other parts of ISO 10303, the SUPERTYPE clause of this entity is only partially explicit. If the subtypes defined in this part of ISO 10303 were a complete categorization, the SUPERTYPE clause of this entity could properly read:

```
SUPERTYPE OF (quantified_assembly_component_usage ANDOR
  ONEOF (next_assembly_usage_occurrence,
         specified_higher_usage_occurrence,
         promissory_usage_occurrence))
```

EXPRESS specification:

```
*)
ENTITY assembly_component_usage
  SUPERTYPE OF (ONEOF (next_assembly_usage_occurrence,
                       specified_higher_usage_occurrence,
                       promissory_usage_occurrence))
  SUBTYPE OF (product_definition_usage);
  reference_designator : OPTIONAL identifier;
END_ENTITY;
(*
```

Attribute definitions:

SELF\product_definition_relationship.relating_product_definition: an assembly for which the related_-product_definition is its constituent.

SELF\product_definition_relationship.related_product_definition: a constituent for which the relating_-product_definition is its parent assembly.

reference_designator: a distinctive code that serves to identify the usage of the related_product_definition as a component in the relating_product_definition in a diagram, list, chart, or on a physical piece of equipment. The reference_designator need not be specified.

NOTE 5 The reference designator attribute can be made mandatory, or constrained to be unique, or both, in an annotated EXPRESS schema that uses or specializes this entity.

4.3.3 assembly_component_usage_substitute

The **assembly_component_usage_substitute** specifies that one constituent can be used as a substitute for another within the context of a given assembly.

NOTE 1 The assembly context is specified by the fact that both **assembly_component_usage**s referred to by the base and substitute attriubtes reference the same **product_definition** in their inherited relating_product_definition attribute.

The instance of the substitute constituent does not require the same spatial relationship or the same quantity. A substitute constituent does not require equivalent form, fit, and function of the constituent for which it is a substitute.

This entity defines one-way substitution only. Within a given context, if A is specified as a substitute for B, B is not an implicit substitute for A.

NOTE 2 The **assembly_component_usage_substitute** entity may be used to eliminate the re-identification of all higher-level assemblies when a new version of a lower-level constituent is created.

EXPRESS specification:

```
*)
ENTITY assembly_component_usage_substitute;
  name              : label;
  definition        : OPTIONAL text;
  base              : assembly_component_usage;
  substitute        : assembly_component_usage;
UNIQUE
  UR1: base,substitute;
WHERE
  WR1: base.relating_product_definition :=:
       substitute.relating_product_definition;
  WR2: base :<>: substitute;
END_ENTITY;
(*
```

Attribute definitions:

name: the **label** by which the **assembly_component_usage_substitute** is known.

definition: the **text** that characterizes the **assembly_component_usage_substitute**. The value of the definition need not be specified.

base: an **assembly_component_usage** for which the substitute may be used.

substitute: an **assembly_component_usage** which may be used for the base.

Formal propositions:

UR1: The combination of the base and substitute attributes shall be unique.

WR1: The relating_product_definition attribute of both the base and the substitute attributes shall refer to the same assembly **product_definition**.

WR2: The base and substitute attributes shall not be the same instance.

4.3.4 assembly_component_usage_substitute_with_ranking

The **assembly_component_usage_substitute_with_ranking** is an **assembly_component_usage_-substitute** that specifies a relative preference value for the usage of a particular constituent as a substitute for another among all substitutes within the context of a given assembly.

EXPRESS specification:

```
*)
ENTITY assembly_component_usage_substitute_with_ranking
  SUBTYPE OF (assembly_component_usage_substitute);
  ranking           : INTEGER;
  ranking_rationale : text;
END_ENTITY;
(*
```

Attribute definitions:

ranking: an integer that ranks the preference for use of the substitute **assembly_component_usage** among all **assembly_component_usage_substitute_with_ranking** instances with the same value for the base attribute. This value is an integer that only has meaning when comparing it with corresponding values for **assembly_component_usage_substitute_with_ranking**s sharing the same base **assembly_component_-usage**. It is a relative ranking value, not an absolute ranking. A lower value indicates a higher preference for the substitute **assembly_component_usage**, and a higher value indicates a lower preference.

ranking_rationale: text that describes the rationale used for the ranking.

EXAMPLE Examples of ranking_rationale are cost and long lead time.

4.3.5 make_from_usage_option

The **make_from_usage_option** is a **product_definition_usage** in which one product is the result of applying a process to another product.

EXAMPLE 1 Machining, plating, and bending are processes used to make another product.

NOTE 1 In situations in which a product is made from another product using a sequence of processes, the intermediate products will be related using the **make_from_usage_option** entity.

NOTE 2 A product to be modified may be an assembly. Generally, the **assembly_component_usage** differs from the **make_from_usage_option** in that the constituents of an assembly are used in the assembly without any change.

NOTE 3 A **product_definition** may be the relating_product_definition of many **make_from_usage_option** relationships, and a **product_definition** may be the related_product_definition of many **make_from_usage_option** relationships. Further, there may be multiple **make_from_usage_option** instances referencing the same relating_-product_definition and related_product_definition pair of **product_definition**s.

EXAMPLE 2 Consider the case of a shaft that can be machined from either a casting or a forging. All three, the shaft, the forging, and the casting, are represented by separate instances of **product_definition**. Two instances of

the **make_from_usage_option** entity exist, one between the relating_product_definition shaft and the related_-product_definition forging, the other between the relating_product_definition shaft and the related_product_-definition casting.

EXPRESS specification:

```
*)
ENTITY make_from_usage_option
  SUBTYPE OF (product_definition_usage);
  ranking           : INTEGER;
  ranking_rationale : text;
  quantity          : measure_with_unit;
WHERE
  WR1: (NOT ('NUMBER' IN TYPEOF(quantity.value_component)))
       OR (quantity.value_component > 0);
END_ENTITY;
(*
```

Attribute definitions:

SELF\product_definition_relationship.relating_product_definition: a **product_definition** made from the related_product_definition product.

SELF\product_definition_relationship.related_product_definition: a **product_definition** from which the relating_product_definition is made.

ranking: an integer that ranks the preference for use of the related_product_definition among all **make_-from_usage_option** instances that have the same value for the inherited relating_product_definition attribute. This value is an integer that only has meaning when comparing it with corresponding values for **make_from_usage_option**s sharing the same relating_product_definition. It is a relative ranking value, not an absolute ranking. A lower value indicates a higher preference for the related_product_definition, and a higher value indicates a lower preference.

ranking_rationale: text that describes the rationale used for the ranking.

EXAMPLE 3 Examples of ranking_rationale are cost and long lead time.

quantity: the amount of the relating_product_definition that can be made from the related_product_-definition.

Formal propositions:

WR1: If the quantity of the constituent is expressed numerically, its value shall be greater than zero.

4.3.6 make_from_usage_option_group

A **make_from_usage_option_group** is a collection of **make_from_usage_option** instances that defines one possible combination of **product_definition**s that can be made from another single **product_-definition**.

NOTE 1 The **make_from_usage_option_group** is used to indicate that several different products may be made from a single product.

The single product constraint is represented by the fact that within a single **make_from_usage_option_-group**, the related_product_definition of all of the instances of the **make_from_usage_option**s shall be the same.

EXAMPLE 1 Suppose a bar stock, represented by **product_definition** D, can be cut twice so as to create three -**product_definition**s, X, Y, and Z. The **make_from_usage_option_group** representing this situation would aggregate three **make_from_usage_option**s as shown in Table 1:

Table 1 - Example 1 make_from_usage_option_group

id	relating_product_-definition	related_product_-definition	quantity
1	X	D	1
2	Y	D	1
3	Z	D	1

NOTE 2 It is possible to use a product to make more than one combination of other products.

EXAMPLE 2 The same bar stock **product_definition** D can be cut to produce two **product_definition**s X and one **product_definition** T. In this case the corresponding **make_from_usage_option_group** would aggregate two **make_from_usage_option**s as shown in Table 2:

Table 2 - Example 2 make_from_usage_option_group

id	relating_product_-definition	related_product_-definition	quantity
2	X	D	2
1	T	D	1

NOTE 3 A single **make_from_usage_option** instance may be part of multiple **make_from_usage_option_group**s.

EXPRESS specification:

*)

```
ENTITY make_from_usage_option_group;
  members : SET [2:?] OF make_from_usage_option;
WHERE
  WR1: SIZEOF (QUERY (example <* members |
       example.related_product_definition
       :=: members[1].related_product_definition)) =SIZEOF(members);
END_ENTITY;
(*
```

Attribute definitions:

members: a collection of at least two **make_from_usage_option** instances whose relating_product_definition instances may be made from the same related_product_definition.

Formal propositions:

WR1: All instances of members shall refer to the same instance of **product_definition** in their respective related_product_definition attribute.

4.3.7 next_assembly_usage_occurrence

The **next_assembly_usage_occurrence** is an **assembly_component_usage** that specifies the relationship between a child constituent and its immediate parent assembly in a product structure.

NOTE 1 An instance of **next_assembly_usage_occurrence** represents an individual occurrence of a constituent in an assembly. Each specific use of the same constituent may be represented by another distinct **next_assembly_usage_occurrence** instance for the purpose of assigning property information such as a position and orientation for the constituent. Property information is not provided by any of the entities of this part of ISO 10303 but by means of using the appropriate capabilities defined in ISO 10303-42 [2], ISO 10303-43 [3], and ISO 10303-41. The representation of a constituent occurrence in an assembly may be linked to mechanisms in ISO 10303-42 [2] and ISO 10303-43 [3] by means of a **property_definition** entity defined in the **product_property_definition_schema** in ISO 10303-41.

EXAMPLE The position and orientation of a constituent with respect to its assembly would be computed using a **transformation** defined in the **representation_schema** in ISO 10303-43 [3].

NOTE 2 An application algorithm can derive an indented parts list for a product by sequentially tracing through a structure of **next_assembly_usage_occurrence** instances. A similar algorithm can be used to calculate the position and orientation of each occurrence of every constituent relative to its higher-level assemblies within a BOM when used in conjunction with the property information constructs defined ISO 10303-41, ISO 10303-42 [2], and ISO 10303-43 [3].

NOTE 3 The number of constituents used in the assembly may be specified by composing a complex instance of **next_assembly_usage_occurrence** and **quantified_assembly_component_usage** (see 4.3.11).

EXPRESS specification:

```
*)
ENTITY next_assembly_usage_occurrence
  SUBTYPE OF (assembly_component_usage);
END_ENTITY;
(*
```


Attribute definitions:

SELF\product_definition_relationship.relating_product_definition: an assembly for which the related_-product_definition is its immediate constituent.

SELF\product_definition_relationship.related_product_definition: a constituent for which the relating_-product_definition is its immediate parent assembly.

4.3.8 product_definition_occurrence_relationship

The **product_definition_occurrence_relationship** is an association between a **product_definition** and an **assembly_component_usage** in which the **product_definition** referenced by the occurrence attribute is a specific instance of the constituent **product_definition** referenced by the related_product_definition attribute of the occurrence_usage in the assembly **product_definition** referenced by the relating_product_-definition attribute.

NOTE The identification of the specific instance of the constituent in the assembly can be defined using the reference_designator attribute of the assembly_component_usage entity.

EXAMPLE - A product structure for a car is defined in which entity instances of **product_definition** are defined for the chassis and the wheel. Four **assembly_component_usage** instances are then created to represent the four wheels of the chassis in which the reference_designator has values of 'left front', 'right front', 'left rear', 'right rear'. To represent specific information about the right rear wheel, a **product_definition** of the wheel is created to which the information can be associated. To specify that this **product_definition** is representing the wheel that is implied by the instance of **assembly_component_usage** with the reference_designator attribute value of 'right rear', the **product_definition_occurence_relationship** is created to associate the **product_definition** with the **assembly_component_usage**.

```
*)
ENTITY product_definition_occurrence_relationship;
  name             : label;
  description      : OPTIONAL text;
  occurrence       : product_definition;
  occurrence_usage : assembly_component_usage;
WHERE
  WR1: occurrence_usage.relating_product_definition :<>:
       occurrence;
  WR2: occurrence_usage.related_product_definition :<>:
       occurrence;
  WR3: occurrence.formation :=:
       occurrence_usage.related_product_definition.formation;
END_ENTITY;
(*
```

Attribute definitions:

name: the **label** by which the **product_definition_occurrence_relationship** is known.

description: the **text** that characterizes the **product_definition_occurrence_relationship**. The value of the description need not be specified.

occurrence: the **product_definition** that represents a specific instance of the product.

occurrence_usage: the **assembly_component_usage** that represents a specific use of the constituent in an assembly.

Formal propositions:

WR1: The **product_definition** referenced by the occurrence attribute shall not be the assembly **product_definition** in the **assembly_component_usage** referenced by the occurrence_usage attribute.

WR2: The **product_definition** referenced by the occurrence attribute shall not be the constituent **product_definition** in the **assembly_component_usage** referenced by the occurrence_usage attribute.

WR3: The **product_definition** referenced by the occurrence attribute and the **product_definition** that represents the constituent of the **assembly_component_usage** shall be definitions of the same **product_definition_formation**.

4.3.9 product_definition_usage

The **product_definition_usage** is a **product_definition_relationship** that specifies a directed association between two **product_definition**s in which the related_product_definition is used in the context of the relating_product_definition. The **product_definition_usage** establishes a relationship stating that the related product is used in the context of the relating product. This usage is established for the life cycle stages and application contexts in which the two products are described.

EXAMPLE 1 The assembly trees established for production sometimes contain additional intermediate levels than the trees established during the design phase. In such a case, the design phase relationship between a component and the assembly in which it is contained may be replaced by several intermediate relationships during production.

EXAMPLE 2 A given product may be used in a support role with respect to another product. A screwdriver may be identified as a resource to be used for the maintenance life cycle stage of a given product. In such a case, the support resource relationship between the screwdriver and the product for which it is identified to be used for maintenance activities may be specified.

EXPRESS specification:

```
*)
ENTITY product_definition_usage
  SUPERTYPE OF (ONEOF (make_from_usage_option,
                       assembly_component_usage))
  SUBTYPE OF (product_definition_relationship);
UNIQUE
  UR1: SELF\product_definition_relationship.id,
       SELF\product_definition_relationship.relating_product_definition,
       SELF\product_definition_relationship.related_product_definition;
WHERE
  WR1:  acyclic_product_definition_relationship
         (SELF,
         [SELF\product_definition_relationship.related_product_definition],
         'PRODUCT_STRUCTURE_SCHEMA.PRODUCT_DEFINITION_USAGE');
END_ENTITY;
(*
```

Attribute definitions:

SELF\product_definition_relationship.id: an identifier for a usage of a **product_definition**. It is used to distinguish between two instances of **product_definition_usage** where the pair of **product_definition** attributes are the same.

EXAMPLE 3 If four identical bolts are used to attach two plates, there may be a need to identify one specific bolt for some purpose. It needs to be torqued to a greater degree than the rest. The inherited id attribute then is used to identify this specific bolt's requirement, even though all four bolt **product_definition_usage**s will have the same attribute pair of **product_definition**s.

Formal propositions:

UR1: The combination of the inherited id, relating_product_definition, and related_product_definition attributes shall be unique.

WR1: Each **product_definition_usage** shall not relate the same relating_product_definition instance of **product_definition** to the related_product_definition or any of its descendants formed by their reference as the relating_product_definition in a graphof **product_definition_usage**s.

NOTE This rule uses the function **acyclic_product_definition_relationship** defined in the **product_definition_-schema** ISO 10303-41.

4.3.10 promissory_usage_occurrence

The **promissory_usage_occurrence** is an **assembly_component_usage** in which the related_product_-definition is in the assembly tree of the relating_product_definition regardless of the number of intermediate levels between them.

NOTE A **promissory_usage_occurrence** is used when the product structure is not completely defined. In such a situation, it is still possible to relate an assembly to a constituent to capture the intent that the constituent will be used in that assembly.

EXPRESS specification:

```
*)
ENTITY promissory_usage_occurrence
  SUBTYPE OF (assembly_component_usage);
END_ENTITY;
(*
```

4.3.11 quantified_assembly_component_usage

The **quantified_assembly_component_usage** is an **assembly_component_usage** that establishes the relationship between an assembly and one of its constituents when there is a need to specify the quantity of the child constituent used in the assembly.

NOTE Generally, for production planning or material planning purposes, several occurrences of a constituent are lumped together, and a quantity is specified to account for the several occurrences. A typical example would be the specifying of an occurrence of a rivet used for joining aeroplane structures and denoting the number of such rivets used on the entire aeroplane. If each of the occurrences of the rivets used is to be specified, then the **next_-assembly_usage_occurrence** entity may be used. As many instances of the **next_assembly_usage_occurrence** as the number of occurrences of the rivets will exist in this case.

EXPRESS specification:

```
*)
ENTITY quantified_assembly_component_usage
  SUBTYPE OF (assembly_component_usage);
  quantity   : measure_with_unit;
WHERE
  WR1: (NOT ('NUMBER' IN TYPEOF(quantity.value_component)))
       OR (quantity.value_component > 0);
END_ENTITY;
(*
```

Attribute definitions:

SELF\product_definition_relationship.relating_product_definition: an assembly for which the related_product_definition is its constituent, and where the quantity of the constituent is specified.

SELF\product_definition_relationship.related_product_definition: a constituent for which the relating_product_definition is its parent assembly, and where the quantity of the constituent is specified.

quantity: a measure that defines how many or how much of the constituent is used in the assembly.

Formal propositions:

WR1: If the quantity of the constituent is expressed numerically, its value shall be greater than zero.

4.3.12 specified_higher_usage_occurrence

The **specified_higher_usage_occurrence** is an **assembly_component_usage** that specifies the relationship between a constituent and an assembly where the assembly is not the immediate parent for the constituent.

The relationship between the constituent and the assembly of the **specified_higher_usage_occurrence** to be specified is captured by the relationship of the inherited attributes relating_product_definition and related_product_definition.

The two attributes (upper_usage and next_usage) within the primary instance of the entity **specified_higher_usage_occurrence** shall respectively specify an **assembly_component_usage** and the **next_assembly_usage_occurrence** which together will provide the definition of the path from the constituent to the assembly for which the **specified_higher_usage_occurrence** is being specified. To ensure that the **next_assembly_usage_occurrence** and the **assembly_component_usage** together constitute the entire path desired for the **specified_higher_usage_occurrence**, it is required that the instance of the related_product_definition attribute of the **assembly_component_usage** entity be the same as the instance of the relating_product_definition attribute of the **next_assembly_usage_occurrence** entity. The related_product_definition attribute of the **next_assembly_usage_occurrence** shall reference the same instance as the related_product_definition attribute of the **specified_higher_usage_occurrence** being specified. The relating_product_definition attribute of the **assembly_component_usage** entity shall reference the same instance as the relating_product_definition attribute of the **specified_higher_usage_occurrence** being specified.

If the **assembly_component_usage** referenced by the attribute upper_usage is not a **next_assembly_-usage_occurrence** it shall be a **specified_higher_usage_occurrence**. This **specified_higher_usage_-occurrence** shall have its attributes upper_usage and next_usage defined as described in the previous paragraph to specify further the path of the primary **specified_higher_usage_occurrence**. This recursive specification shall continue until the attribute upper_usage references an **assembly_component_usage** entity that is a **next_assembly_usage_occurrence**. At this point, the primary **specified_higher_usage_-occurrence** is fully specified both in terms of its constituents/assembly relationship and the entire path between them.

NOTE 1 The **specified_higher_usage_occurrence** is used to define portions of parts lists that contain a specific constituent within an assembly for which certain properties are to be associated or when the portion is used in more than one assembly structure.

NOTE 2 Annex E contains examples of this entity and figures that depict the relationships between it and the other entities of this schema.

EXPRESS specification:

```
*)
ENTITY specified_higher_usage_occurrence
  SUBTYPE OF (assembly_component_usage);
  upper_usage   : assembly_component_usage;
  next_usage    : next_assembly_usage_occurrence;
UNIQUE
  UR1: upper_usage, next_usage;
WHERE
  WR1: SELF :<>: upper_usage;
  WR2: SELF\product_definition_relationship.relating_product_definition
       :=: upper_usage.relating_product_definition;
  WR3: SELF\product_definition_relationship.related_product_definition
       :=: next_usage.related_product_definition;
  WR4: (upper_usage.related_product_definition :=:
       next_usage.relating_product_definition) OR
       (SIZEOF (QUERY (pdr <* USEDIN
       (upper_usage.related_product_definition,
       'PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION_RELATIONSHIP.' +
       'RELATED_PRODUCT_DEFINITION') |
        pdr.relating_product_definition :=:
        next_usage.relating_product_definition)) = 1);
  WR5: SIZEOF (['PRODUCT_STRUCTURE_SCHEMA.NEXT_ASSEMBLY_USAGE_OCCURRENCE',
       'PRODUCT_STRUCTURE_SCHEMA.SPECIFIED_HIGHER_USAGE_OCCURRENCE']
       * TYPEOF(upper_usage)) = 1;
END_ENTITY;
(*
```

Attribute definitions:

SELF\product_definition_relationship.relating_product_definition: the inherited attribute for the assembly product definition of the **specified_higher_usage_occurrence**.

SELF\product_definition_relationship.related_product_definition: the inherited attribute for the constituent product definition of the **specified_higher_usage_occurrence**.

upper_usage: an **assembly_component_usage** that is the same instance of the attribute relating_product_definition as this **specified_higher_usage_occurrence** and the same instance of the attribute related_product_definition as the relating_product_definition of the **next_assembly_usage_occurrence** referenced by the attribute next_usage.

next_usage: a **next_assembly_usage_occurrence** that is the same instance of the attribute related_product_definition as this **specified_higher_usage_occurrence** and the same instance of the product definition referenced by the attribute relating_product_definition as the product definition referenced by the attribute related_product_definition of the attribute upper_usage.

NOTE 3 See Figure E.6 for diagrams illustrating instances of these two attributes.

Formal propositions:

UR1: The combination of the upper_usage and next_usage attributes shall be unique.

WR1: The instance of **specified_higher_usage_occurrence** shall not be the same as the instance of upper_usage.

WR2: The relating_product_definition attribute (i.e., assembly) of the **specified_higher_usage_occurrence** shall reference the same instance of **product_definition** as the relating_product_definition (i.e., assembly) for the upper_usage.

WR3: The related_product_definition attribute (i.e., constituent) of the **specified_higher_usage_occurrence** shall reference the same instance of **product_definition** as the related_product_definition for the next_usage.

WR4: The related_product_definition (i.e., constituent) for the upper_usage shall reference, or be related to, the same instance of **product_definition** as the relating_product_definition (i.e., assembly) for the next_usage.

WR5: The type of the upper_usage attribute shall be either **next_assembly_usage_occurrence** or **specified_higher_usage_occurrence**.

```
*)
END_SCHEMA; -- product_structure
(*
```

5 Product concept

The following EXPRESS declaration begins the **product_concept_schema** and identifies the necessary external references.

EXPRESS specification:

```
*)
SCHEMA product_concept_schema;
```



```
  REFERENCE FROM support_resource_schema
    (text, label, identifier);

  REFERENCE FROM application_context_schema
    (product_concept_context);
(*
```

NOTE 1 The schemas referenced above are specified in the following part of ISO 10303:

support_resource_schema	ISO 10303-41
application_context_schema	ISO 10303-41

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

5.1 Introduction

The subject of the **product_concept_schema** is the idea of a product as defined by customer needs, i.e., a product concept. A product concept may exist before a product has been defined. A product concept identifies a selection of product features or capabilities. A product concept may also have product features identified for it. Conditions among those features for a particular product may be specified to manage dependencies and define the valid product variations for a particular product concept.

EXAMPLE An automobile may have two types of engines identified as product features. One is a low power engine; the other is a high power engine. Air conditioning may also be identified as a product feature for the same automobile. A condition would be specified that when the air conditioning is present, the high power engine is necessary.

5.2 Fundamental concepts and assumptions

The following concepts and assumptions apply.

— A product concept identifies a deliverable product defined by a producer to satisfy a customer. A product concept is often used to identify a selection of product features or capabilities.

— A product concept may have several configuration items identified for it.

— A product concept may exist without a product or product version.

— A product concept may have product features identified for it defined by a producer to satisfy a customer viewpoint.

— A set of rules for required interdependent combinations of a product feature may be specified.

NOTE Examples and diagrams of the manner in which the entities of this part of ISO 10303 may be used to represent the combinations of product features are included in annex E.

5.3 Product concept entity definitions

5.3.1 concept_feature_operator

A **concept_feature_operator** defines an operation that is used to constrain the relationship between two **product_concept_feature**s.

EXAMPLE "AND", "NOT", and "OR" are **concept_feature_operator** names.

NOTE 1 The meaning of the operation and the list of allowed values for the name attribute can be specified in each annotated EXPRESS schema that uses or specializes this entity or in an agreement of common understanding between the partners sharing this information.

NOTE 2 Annex E contains examples of this entity and figures that depict the relationships between it and the other entities of this schema.

EXPRESS specification:

```
*)
ENTITY concept_feature_operator;
  name        : label;
  description : OPTIONAL text;
END_ENTITY;
(*
```

Attribute definitions:

name: the **label** by which the **concept_feature_operator** is known.

NOTE 3 The name designates the operation.

description: the **text** that characterizes the **concept_feature_operator**. The value of the description need not be specified.

5.3.2 concept_feature_relationship

A **concept_feature_relationship** is an association between two **product_concept_feature**s.

NOTE The meaning of the association can be specified in each annotated EXPRESS schema that uses or specializes this entity or in an agreement of common understanding between the partners sharing this information.

EXPRESS specification:

```
*)
ENTITY concept_feature_relationship;
  name                            : label;
  description                     : OPTIONAL text;
  relating_product_concept_feature  : product_concept_feature;
  related_product_concept_feature   : product_concept_feature;
END_ENTITY;
(*
```

Attribute definitions:

name: the **label** by which the **product_concept_feature_relationship** is known.

description: the **text** that characterizes the **product_concept_feature_relationship**. The value of the description need not be specified.

relating_product_concept_feature: one of the **product_concept_feature**s that is a part of the association.

related_product_concept_feature: the other **product_concept_feature** that is a part of the association. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

5.3.3 concept_feature_relationship_with_condition

A **concept_feature_relationship_with_condition** is a **concept_feature_relationship** in which a conditional operator characterizes the relationship between the related_product_concept_feature and the relating_product_concept_feature.

NOTE 1 When specified as the condition for a **conditional_concept_feature**, a **concept_feature_relationship_with_condition** defines a constraint for the use of a **product_concept_feature**. Complex conditions can be constructed for a **product_concept_feature** by using the **concept_feature_relationship_with_condition** to relate **conditional_concept_feature**s to each other.

NOTE 2 Annex E contains examples of this entity and figures that depict the relationships between it and the other entities of this schema.

EXPRESS specification:

```
*)
ENTITY concept_feature_relationship_with_condition
  SUBTYPE OF (concept_feature_relationship);
  conditional_operator : concept_feature_operator;
END_ENTITY;
(*
```

Attribute definitions:

conditional_operator: the **concept_feature_operator** used in the relationship between the relating_product_concept_feature and the related_product_concept_feature.

EXAMPLE A condition of negation is specified by using a **concept_feature_operator** with a name of "NOT" and a **product_concept_feature_relationship_with_condition** in which the related_concept_feature and the relating_concept_feature reference the same instance of **product_concept_feature**.

5.3.4 conditional_concept_feature

A **conditional_concept_feature** is a **product_concept_feature** that is the combination of two **product_concept_feature**s constrained by an operator.

NOTE 1 Complex combinations of **product_concept_feature**s may be constructed using **conditional_concept_feature**s as operands of a **concept_feature_relationship_with_condition**.

NOTE 2 Annex E contains examples of this entity and figures that depict the relationships between it and the other entities of this schema.

EXPRESS specification:

```
*)
ENTITY conditional_concept_feature
  SUBTYPE OF (product_concept_feature);
  condition : concept_feature_relationship_with_condition;
END_ENTITY;
(*
```

Attribute definitions:

condition: the **concept_feature_relationship_with_condition** that establishes the condition for the **conditional_concept_feature**.

5.3.5 product_concept

A **product_concept** is is a type of product as spefied in 3.3.4 of ISO 10303-41 that is defined by a producer to satisfy potential or actual customer requirements.

NOTE 1 A **product_concept** will often correspond to the highest level item(s) manufactured by an organization for a customer. It may be characterized by a set of product features identified by the customers or derived from customers' needs. The definition of product concepts is often driven by market and customer requirements and forecasting.

EXAMPLE 1 If an organization manufactures cars and engines for cars, the cars will be represented by **product_concept** instances. If another organization manufactures engines for cars, then the engines will be represented as **product_concept**s in that organization.

EXPRESS specification:

```
*)
ENTITY product_concept;
  id                  : identifier;
  name                : label;
  description         : OPTIONAL text;
  market_context      : product_concept_context;
UNIQUE
  UR1: id;
END_ENTITY;
(*
```

Attribute definitions:

id: the **identifier** that distinguishes the **product_concept**.

EXAMPLE 2 The id could be a sales model number.

name: the **label** by which the **product_concept** is known.

description: the **text** that characterizes the **product_concept**. The value of the description need not be specified.

NOTE 2 This attribute can be used to convey the purpose, functionality, and selected features for the **product_concept**.

market_context: the frame of reference in which the **product_concept** is defined.

NOTE 3 The name of the market_context attribute is being maintained due to requirements for upwardly compatable revisions to this part of ISO 10303. However, the frame of reference for the product_concept is not restricted to a market context.

NOTE 4 The value of the market_context attribute can be constrained in each annotated EXPRESS schema that uses or specializes this entity data type or in an agreement of common understanding between the partners sharing this information.

Formal propositions:

UR1: The value of the id attribute shall be unique.

5.3.6 product_concept_feature

A **product_concept_feature** identifies a characteristic intended to be used to distinguish variations among the products of one or more **product_concept**s.

EXAMPLE "Luxury equipment" is a characteristic that distinguishes between a standard, low-price car and a top-class car.

NOTE The categorization of **product_concept_feature**s is accomplished through the mechanisms defined in ISO 10303-41 for collecting product data into groups. The name of the group indicates the category of the collected items.

EXPRESS specification:

```
*)
ENTITY product_concept_feature;
  id          : identifier;
  name        : label;
  description : OPTIONAL text;
END_ENTITY;
(*
```

Attribute definitions:

id: the **identifier** that distinguishes the **product_concept_feature**.

name: the **label** by which the **product_concept_feature** is known.

description: the **text** that characterizes the **product_concept_feature**. The value of the description need not be specified.

5.3.7 product_concept_feature_association

The **product_concept_feature_association** relates a **product_concept_feature** to a **product_concept**. A **product_concept_feature** as it is related to a **product_concept** may define the content of a **configuration_item**.

NOTE The meaning of the association can be specified in each annotated EXPRESS schema that uses or specializes this entity or in an agreement of common understanding between the partners sharing this information.

EXPRESS specification:

```
*)
ENTITY product_concept_feature_association;
  name        : label;
  description : OPTIONAL text;
  concept     : product_concept;
  feature     : product_concept_feature;
END_ENTITY;
(*
```

Attribute definitions:

name: the **label** by which the **product_concept_feature_association** is known.

description: the **text** that characterizes the **product_concept_feature_association**. The value of the description need not be specified.

concept: the **product_concept** that is being associated.

feature: the **product_concept_feature** that is being associated.

5.3.8 product_concept_relationship

A **product_concept_relationship** is an association between two **product_concept**s.

NOTE The meaning of the association can be specified in each annotated EXPRESS schema that uses or specializes this entity or in an agreement of common understanding between the partners sharing this information.

EXPRESS specification:

```
*)
ENTITY product_concept_relationship;
  name                    : label;
  description             : OPTIONAL text;
  relating_product_concept : product_concept;
  related_product_concept  : product_concept;
END_ENTITY;
(*
```

Attribute definitions:

name: the **label** by which the **product_concept_relationship** is known.

description: the **text** that characterizes the **product_concept_relationship**. The value of the description need not be specified.

relating_product_concept: one of the **product_concept**s that is a part of the association.

related_product_concept: the other **product_concept** that is a part of the association. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

```
*)
END_SCHEMA; -- product_concept_schema
(*
```

6 Configuration management

The following EXPRESS declaration begins the **configuration_management_schema** and identifies the necessary references.

EXPRESS specification:

```
*)
SCHEMA configuration_management_schema;

  REFERENCE FROM product_definition_schema
    (product_definition,
     product_definition_formation,
     product_definition_effectivity);

  REFERENCE FROM product_property_representation_schema
    (relatives_of_product_definitions);

  REFERENCE FROM product_structure_schema
    (product_definition_usage);

  REFERENCE FROM product_concept_schema
    (product_concept,
     product_concept_feature_association);

  REFERENCE FROM basic_attribute_schema
    (get_description_value,
     get_name_value);

  REFERENCE FROM support_resource_schema
    (text, label, identifier, bag_to_set);
(*
```

NOTE 1 The schemas referenced above are specified in the following parts of ISO 10303:

product_definition_schema	ISO 10303-41
product_property_representation_schema	ISO 10303-41
product_structure_schema	clause 4 of this part of ISO 10303
product_concept_schema	clause 5 of this part of ISO 10303

basic_attribute_schema	ISO 10303-41
support_resource_schema	ISO 10303-41

NOTE 2 See annex D for a graphical presentation of this schema using the EXPRESS-G notation.

NOTE 3 A listing of the complete EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is available from the Internet - see annex C.

6.1 Introduction

The subject of the **configuration_management_schema** is the identification of items, the composition of which is to be managed. The item to be managed is specified as a **configuration_item**. It is usually visible to customers of the organization that does the configuration management. If the item being managed is a product, the schema facilitates the establishment of the association of appropriate versions of the product to the **configuration_item**.

This schema also provides the capability to record the associations established to enable the tracking of the associations.

The functionality of configuration management is achieved using the entities **configuration_item**, **configuration_design**, and **configuration_effectivity**.

The configuration management schema of this part of ISO 10303 contains the following concepts:

— The identification of the **configuration_item**s and the respective **product_concept**s of which they form a constituent.

— The identification of a **product_definition** or **product_definition_formation** that realizes the **configuration_item**.

— The association of the appropriate versions of a product to build a **configuration_item**. This association is referred to as **configuration_effectivity**.

— Three of the ways to apply **configuration_effectivity** are:

a) **serial_numbered_effectivity**, where the **configuration_effectivity** is based on serial numbered instances of manufactured products;

b) **dated_effectivity**, where the **configuration_effectivity** is based on dates when instances of the product are manufactured;

c) **lot_effectivity**, where the **configuration_effectivity** is based on instances of lots of products manufactured.

NOTE The definitions of **serial_numbered_effectivity**, **dated_effectivity**, and **lot_effectivity** are given in the **effectivity_schema** in ISO 10303-41.

6.2 Fundamental concepts and assumptions

The following concepts and assumptions apply.

— Configuration management may be applied to products and parts of products to be designed or manufactured;

— An organization determines which products are to be under its configuration management control. These products become the configuration items of the organization. These usually are higher-level functional elements that act as the focal points for managing the effectivity of lower-level constituents;

— A product concept may identify a customer view of a product based on a static set of configuration items or may be configurable based on customer selected options that are defined for it.

6.3 Configuration management type definition: configuration_design_item

The **configuration_design_item** type represents the design that is a candidate for use in manufacturing actual units associated with a **configuration_item**.

EXPRESS specification:

```
*)
TYPE configuration_design_item = SELECT
   (product_definition,
    product_definition_formation);
END_TYPE;
(*
```

6.4 Configuration management entity definitions

6.4.1 configurable_item

A **configurable_item** is a **configuration_item** that is characterized by a set of **product_concept_feature**s that have been identified for use in a **product_concept**. **Product_concept**s specified by the **product_-concept_feature_association**s shall be the same as or related to the one referenced by the inherited item_-concept attribute.

EXAMPLE A customer order for a car specifies the model of car ordered and also certain features of the car, such as colour "red", trim "leather grey", "sun roof", and "aluminium rims". The car that is ordered can be represented by a **configurable_item**. The model of car can be represented by the **product_concept** that plays the role item_-concept for the ordered car. Each of the features of the ordered car can be represented by a **product_concept_-feature**.

EXPRESS specification:

```
*)
ENTITY configurable_item
  SUBTYPE OF (configuration_item);
  item_concept_feature : SET[1:?] OF product_concept_feature_association;
END_ENTITY;
(*
```

Attribute definitions:

item_concept_feature: the **product_concept_feature_association** instances associated with the **configurable_item**.

6.4.2 configuration_design

The **configuration_design** relates a configuration item and a product design intended to implement that item. Thus, a **configuration_design** entity represents the association of a **configuration_item** with a **product_definition** or **product_definition_formation** to specify that the corresponding design is an element of a solution for a given **configuration_item**.

NOTE 1 Organizations may establish this association before any actual units are planned and before any details of the design have been established.

EXPRESS specification:

```
*)
ENTITY configuration_design;
  configuration : configuration_item;
  design        : configuration_design_item;
DERIVE
  name          : label := get_name_value (SELF);
  description   : text := get_description_value (SELF);
UNIQUE
  UR1: configuration, design;
WHERE
  WR1: SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                             'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
  WR2: SIZEOF (USEDIN (SELF, 'BASIC_ATTRIBUTE_SCHEMA.' +
                             'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY;
(*
```

Attribute definitions:

configuration: a **configuration_item** that specifies that the **product_definition** or **product_definition_-formation** is a candidate to be treated as a single unit in the configuration management process.

design: a **product_definition** or **product_definition_formation** representing a design that is a candidate to be treated as a single unit in the configuration management process.

name: the **label** by which the **configuration_design** is known.

NOTE 2 This attribute is an upwardly compatible addition to **configuration_design** as specified in ISO 10303-44:1994.

description: the **text** that characterizes the **configuration_design**. The value of the description need not be specified.

NOTE 3 This attribute is an upwardly compatible addition to **configuration_design** as specified in ISO 10303-44:1994.

Formal propositions:

UR1: The combination of the configuration attribute and the design attribute shall be unique.

WR1: Each **configuration_design** shall be the named_item in at most one **name_attribute**.

NOTE 4 The **name_attribute** data type is defined in the **basic_attribute_schema** in ISO 10303-41.

WR2: Each **configuration_design** shall be the described_item in at most one **description_attribute**.

NOTE 5 The **description_attribute** data type is defined in the **basic_attribute_schema** in ISO 10303-41.

NOTE 6 a template for constraining the population of the entity data types defined in the basic_attribute_schema is described in Annex E of ISO 10303-41.

6.4.3 configuration_effectivity

A **configuration_effectivity** is the identification of a valid use of a particular **product_definition** in the context of its participation, as the related_product_definition, in a **product_definition_usage** as an element of a solution for a given **configuration_item**. The solution for a **configuration_item** is specified by the **configuration_design** given in the configuration attribute.

EXAMPLE A fuel injection pump in a 200 HP engine is effective for a certain date range in a particular car. This information is captured prior to any production plans existing for the 200 HP engine in a **configuration_effectivity** entity.

NOTE The **product_definition_usage** data type is a subtype of the **effectivity** data type. Other subtypes of **effectivity** are defined in the **effectivity_schema** in ISO 10303-41. An instance of this entity may also be an instance of either a **serial_numbered_effectivity**, a **dated_effectivity**, or a **lot_effectivity**.

In the case where the applicability of the **product_definition_usage** is defined using **serial_numbered_-effectivity** or **lot_effectivity**, the product that provides the context for the serial number or the lot number is specified by the **configuration_design**.

EXPRESS specification:

```
*)
ENTITY configuration_effectivity
  SUBTYPE OF (product_definition_effectivity);
  configuration : configuration_design;
UNIQUE
  UR1: configuration,
       usage,
       id;
WHERE
  WR1: 'PRODUCT_STRUCTURE_SCHEMA.PRODUCT_DEFINITION_USAGE' IN
        TYPEOF (SELF\product_definition_effectivity.usage);
END_ENTITY;
(*
```

Attribute definitions:

configuration: a **configuration_design** for which the **configuration_effectivity** applies.

Formal propositions:

UR1: The combination of the configuration attribute, the usage attribute, and the id attribute shall be unique.

WR1: The usage shall be a **product_definition_usage**.

Informal propositions:

IP1: The relating_product_definition of the **product_definition_usage** referenced by the usage attribute shall be the same instance of **product_definition** as the one referenced by the design attribute of the **configuration_design**, or it shall be related to that instance in a tree of **assembly_component_usage** instances.

6.4.4 configuration_item

A **configuration_item** is the identification of an idea of a product or any of the discrete portions of a product, that is designated for configuration management and treated as a single unit in the configuration management process.

NOTE 1 A **configuration_item** may be a variation of a **product_concept**, an entire **product_concept**, or some portion thereof.

NOTE 2 A **configuration_item** can be established prior to the existence of a corresponding **product_definition** or **product_definition_formation**.

NOTE 3 The association between a **configuration_item** and a corresponding **product_definition** or **product_definition_formation** is established using a **configuration_design**.

EXPRESS specification:

```
*)
ENTITY configuration_item;
   id                  :  identifier;
   name                :  label;
   description         :  OPTIONAL text;
   item_concept        :  product_concept;
   purpose             :  OPTIONAL label;
END_ENTITY;
(*
```

Attribute definitions:

id: the **identifier** that distinguishes the **configuration_item**.

name: the **label** by which the **configuration_item** is known.

description: the **text** that characterizes the **configuration_item**. The value of the description need not be specified.

NOTE 4 The description may identify the specification and function of the **configuration_item**.

item_concept: a **product_concept** associated with the **configuration_item**.

purpose: a descriptive label providing a reason to create the item_concept. The purpose need not be specified.

6.4.5 configuration_item_relationship

A **configuration_item_relationship** is an association between two **configuration_item**s.

NOTE The meaning of the association can be specified in each annotated EXPRESS schema that uses or specializes this entity or in an agreement of common understanding between the partners sharing this information.

EXPRESS specification:

```
*)
ENTITY configuration_item_relationship;
  name                        : label;
  description                 : OPTIONAL text;
  relating_configuration_item : configuration_item;
  related_configuration_item  : configuration_item;
END_ENTITY;
(*
```

Attribute definitions:

name: the **label** by which the **configuration_item_relationship** is known.

description: the **text** that characterizes the **configuration_item_relationship**. The value of the description need not be specified.

relating_configuration_item: one of the **configuration_item**s that is a part of the association.

related_configuration_item: the other **configuration_item** that is a part of the association. If one element of the relationship is dependent upon the other, this attribute shall be the dependent one.

```
*)
END_SCHEMA; -- configuration_management_schema
(*
```

Annex A
(normative)

Short names of entities

Table A.1 provides the short names of entities specified in this part of ISO 10303. Requirements on the use of the short names are found in the implementation methods included in ISO 10303.

NOTE The EXPRESS entity names are available from Internet:

<http://www.mel.nist.gov/div826/subject/apde/snr/>.

Table A.1 - Short names of entities

Entity names	Short names
alternate_product_relationship	ALPRRL
assembly_component_usage	ASCMUS
assembly_component_usage_substitute	ACUS
assembly_component_usage_substitute_with_ranking	ACUSWR
concept_feature_operator	CNFTOP
concept_feature_relationship	CNFTRL
concept_feature_relationship_with_condition	CFRWC
conditional_concept_feature	CNCNFT
configurable_item	CNF0
configuration_design	CNFDSG
configuration_effectivity	CNFEFF
configuration_item	CNFITM
configuration_item_relationship	CNITRL
make_from_usage_option	MFUO
make_from_usage_option_group	MFUOG
next_assembly_usage_occurrence	NAUO
product_concept	PRDCNC
product_concept_feature	PRCNFT
product_concept_feature_association	PCFA
product_concept_relationship	PRCNRL

Table A.1 - Short names of entities (concluded)

Entity names	Short names
product_definition_occurrence_relationship	PRDFOR
product_definition_usage	PRDFUS
promissory_usage_occurrence	PRUSOC
quantified_assembly_component_usage	QACU
specified_higher_usage_occurrence	SHUO

Annex B
(normative)

Information object registration

B.1 Document identification

To provide for unambiguous identification of an information object in an open system, the object identifier

{ iso standard 10303 part(44) version(3) }

is assigned to this part of ISO 10303. The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2 Schema identification

B.2.1 product_structure_schema identification

To provide for unambiguous identification of the **product_structure_schema** in an open information system, the object identifier

{ iso standard 10303 part(44) version(3) object(1) product-structure-schema(1) }

is assigned to the **product_structure_schema** schema (see clause 4). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.2 product_concept_schema identification

To provide for unambiguous identification of the **product_concept_schema** in an open information system, the object identifier

{ iso standard 10303 part(44) version(2) object(1) product-concept-schema(2) }

is assigned to the **product_concept_schema** schema (see clause 5). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

B.2.3 configuration_management_schema identification

To provide for unambiguous identification of the **configuration_management_schema** in an open information system, the object identifier

{ iso standard 10303 part(44) version(2) object(1)
configuration-management-schema(3) }

is assigned to the **configuration_management_schema** schema (see clause 6). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

Annex C
(informative)

Computer-interpretable listing

This annex references a listing of the EXPRESS entity names and corresponding short names as specified in this part of ISO 10303. It also references a listing of each EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text. These listings are available in computer-interpretable form and can be found at the following URLs:

Short names: http://www.mel.nist.gov/div826/subject/apde/snr/
EXPRESS: http://www.mel.nist.gov/step/parts/part044e2/dis/

If there is difficulty accessing these sites contact ISO Central Secretariat or contact the ISO TC 184/SC4 Secretariat directly at: sc4sec@cme.nist.gov.

NOTE The information provided in computer-interpretable form at the above URLs is informative. The information that is contained in the body of this part of ISO 10303 is normative.

Annex D
(informative)

EXPRESS-G diagrams

Figures D.1 to D.3 correspond to the EXPRESS listing given in clauses 4 to 6. The diagrams use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in annex D of ISO 10303-11.

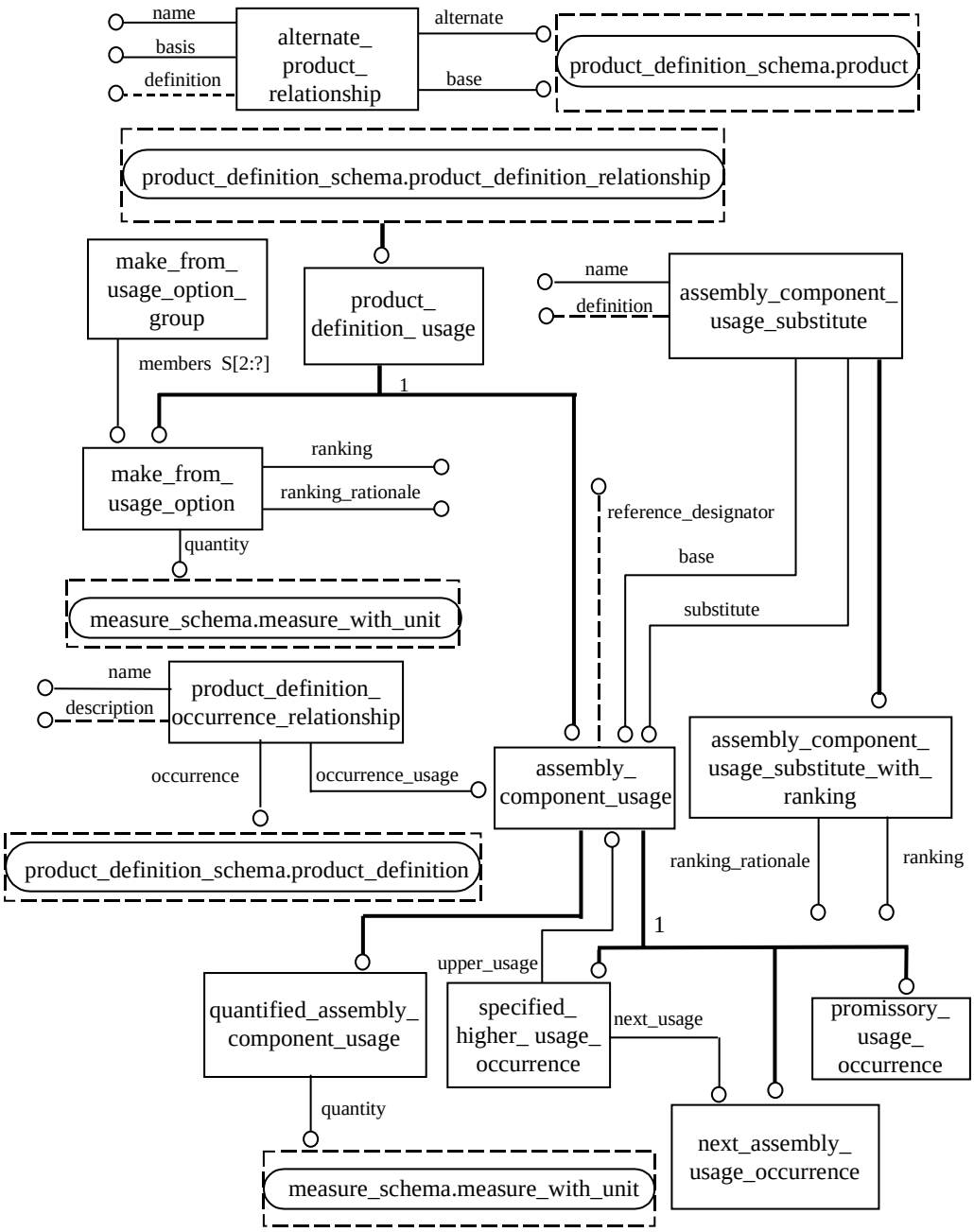

Figure D.1 - product_structure_schema

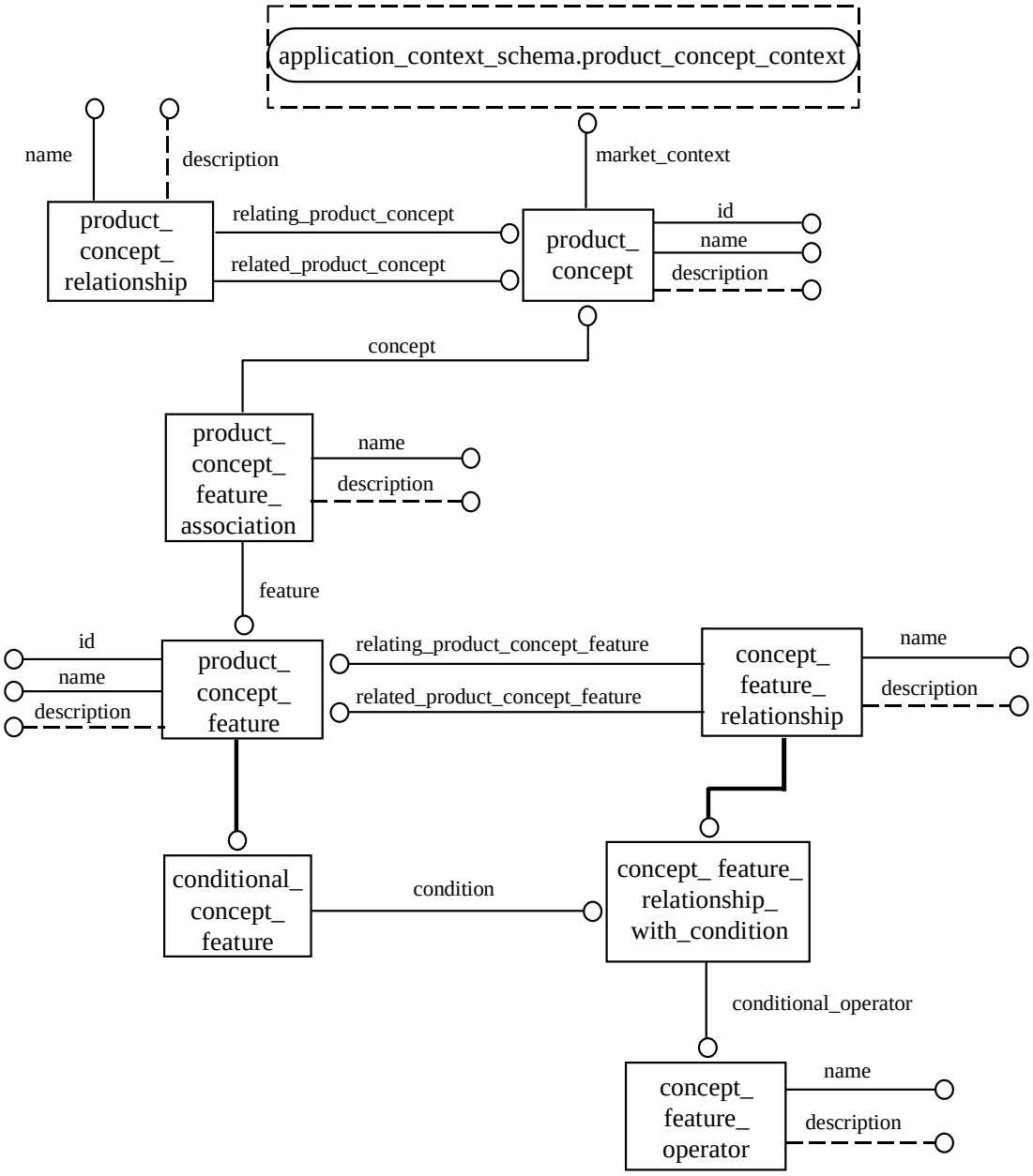

Figure D.2 - product_concept_schema

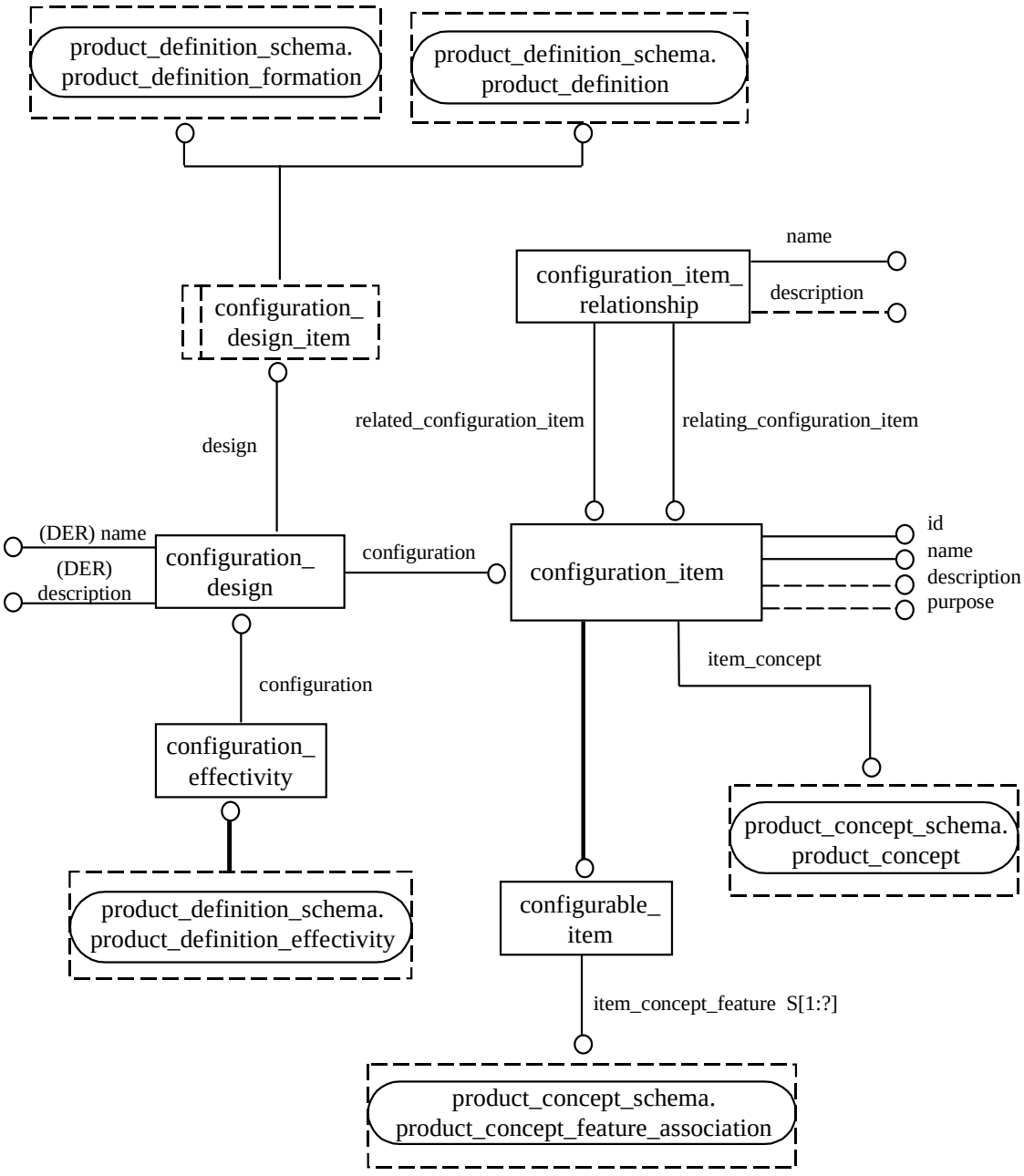

Figure D.3 - configuration_management_schema

Annex E
(informative)

Examples

E.1 Product structure

Product structure reports are used by industry to describe the composition and structure of a product. Most products are assembled from other assemblies and components. A product structure report provides details about what subassemblies and components are used.

NOTE See definitions in clause 3.

In this annex, examples of three different types of product structure reports are provided. In addition, the use of this part of ISO 10303 to generate these reports is outlined. The focus of the examples is on the **product_structure_schema**.

E.1.1 Example of a mechanical assembly product

In this section, a hypothetical mechanical assembly product called the "part 44 example product" is described, both in diagrams and in words. This assembly forms the basis for explaining different types of product structure reports, e.g., BOM and parts list, in the next section. This product is modelled using the various entities defined earlier in this part of ISO 10303. See Figure E.2 for the 3D drawing of the "part 44 example product", and Figure E.2 for a diagrammatic representation of the product structure.

The "part 44 example product" represents a simple mechanical assembly consisting of three major constituents: a large metal plate forming the base for attaching two identical L-bracket assemblies. Each of the two L-bracket assemblies is bolted to the base using three nuts and bolts and using three holes drilled into the plate. The L-brackets themselves have a hole drilled in them to support a rod assembly that consists of a threaded rod and nuts attached at each end of the rod. The nuts used on the threaded rod and in the L-bracket assembly are of the same size. This product design has a structure consisting of a number of assemblies, some of which also contain assemblies, i.e., sub-assemblies, as well as a number of individual components.

E.1.2 Examples of product structure reports

Product structure reports can decompose an assembly to many levels of details. The levels of detail can cover the degree of decomposition, the quantities of subassemblies and components used, the position and orientation of the subassemblies and components, and the products used to make other products. The levels of detail include:

— degree of decomposition;

  a) single-level: a single assembly and its immediate constituent sub-assemblies or components;

  b) multi-level: one or more assemblies in which at least one subassembly is further decomposed;

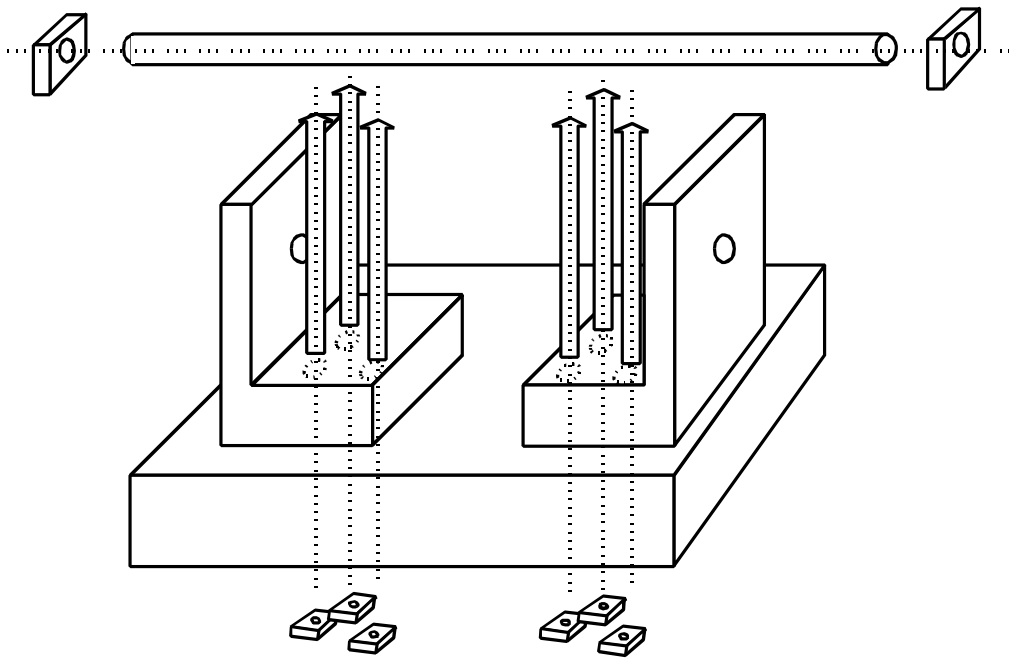

Figure E.1 - Part 44 example product

— type of decomposition;

a) extended definition: the products used to make one or more components are identified;

b) exploded definition: for a given level of decomposition, each kind of subassembly and component is described; multiple uses of the same kind of subassembly or component are not described;

c) flattened: one assembly described in terms of its components; no subassemblies are described;

d) numeric: for a given level of decomposition, each subassembly and component is described only once with its quantity or amount;

e) exploded occurrence: for a given level of decomposition, every subassembly and component is described, even if it is identical with another subassembly or component;

f) labelled occurrence: an exploded occurrence decomposition in which each subassembly and component is labelled;

g) positioned: each constituent is associated with its position and orientation relative to the coordinate system of its immediate assembly;

h) exploded hybrid: for a given level of decomposition, each subassembly and component is described; subassemblies or components may be described by an exploded definition or an exploded occurrence.

— style of decomposition: each level of decomposition may be indented with respect to the previous level.

Examples of typical reports having a mixture of the characteristics enumerated above are now presented using the "part 44 example product".

E.1.2.1 Example: A numeric, multi-level, indented product structure report

Each assembly, subassembly, and component is placed on a new line. A numeric value, indicating the count or amount of the constituent, may be present. The nesting of the levels is indicated by indentation. Constituents at the same level have the same indentation. This report shows that the "part 44 example product" has two L-bracket assembly units and that each unit contains three nut-bolt assemblies. The rod assembly contains two nuts and a rod.

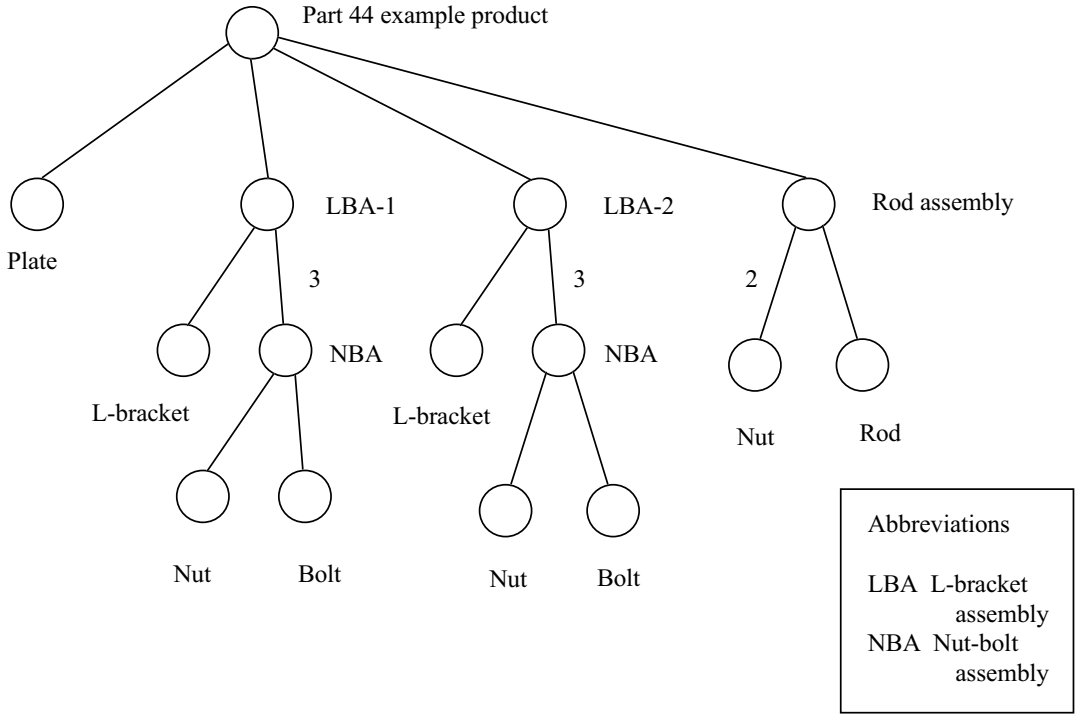

Figure E.2 - Part 44 example product: graphical product structure

Part 44 Example Product
    Plate
    L-bracket assembly (2)
        L-bracket
        Nut-bolt assembly (3)
            Nut
            Bolt

Rod assembly
Rod
Nut (2)

E.1.2.2 Example: A labelled occurrence, multi-level, indented product structure report

In this report, the structure is decomposed to the lowest level of the product structure. A label is assigned to each subassembly and component, down to the lowest level, e.g., Nut-1-1. With this form, all subassemblies and components can be referenced uniquely within the complete product structure and can be distinguished from one another. One can retrieve Nut-2-1 (the first nut on the second L-bracket assembly) and differentiate it from Nut-rod-2 (the second nut on the rod assembly).

Part 44 Example Product
Plate-1: Plate
L-bracket-assembly-1: L-bracket assembly
L-bracket-1-1: L-bracket
Nut-bolt-assembly-1-1: Nut-bolt assembly
Nut-1-1: Nut
Bolt-1-1: Bolt
Nut-bolt-assembly-1-2: Nut-bolt assembly
Nut-1-2: Nut
Bolt-1-2: Bolt
Nut-bolt-assembly-1-3: Nut-bolt assembly
Nut-1-3: Nut
Bolt-1-3: Bolt
L-bracket-assembly-2: L-bracket assembly
L-bracket-2-1: L-bracket
Nut-bolt-assembly-2-1: Nut-bolt assembly
Nut-2-1: Nut
Bolt-2-1: Bolt
Nut-bolt-assembly-2-2: Nut-bolt assembly
Nut-2-2: Nut
Bolt-2-2: Bolt
Nut-bolt-assembly-2-3: Nut-bolt assembly
Nut-2-3: Nut
Bolt-2-3: Bolt
Rod-assembly-1: Rod assembly
Rod-1: Rod
Nut-Rod-1: Nut
Nut-Rod-2: Nut

E.1.2.3 Example: A numeric, multi-level, extended definition, indented product structure report

This report includes the number of each constituent. This report is an extended definition because it references the product from which some of the components are made; e.g., the sheet metal used to make the plate.

Part 44 Example Product
    Plate
        Sheet metal
    L-bracket assembly (2)
        L-bracket
            Sheet metal
        Nut-bolt assembly (3)
            Nut
            Bolt
    Rod assembly
        Rod
            Bar stock
        Nut (2)

E.1.3 BOM, parts list, and hybrid data structures

In order for a compliant application to generate the information normally expected in the various product structure reports in the last section, it will need the information to be represented in data structures for physical storage. This example introduces three data structures that may be used to represent this information. They are the BOM data structure, the parts list data structure, and the hybrid data structure that are defined in E.1.3.1, E.1.3.2, and E.1.3.3.

E.1.3.1 BOM data structures

The BOM data structure is used to represent the assembly aspects of product structure. The BOM structure specifies only the different products used in an assembly, and thus provides a list of products necessary to create the assembly. Each of these products, as identified in the structure, is a typical product and is called a definition. Only one definition is included for each kind of product that participates in the structure. Although there are eight nuts specified in the "part 44 example product" above, the BOM data structure representing this structure would have only one nut to represent that kind of product in the "part 44 example product". (See Figure E.3)

The BOM data structure does support the representation of multiple relationships between two specific definitions. To indicate that there are three nut-bolt assemblies in the "part 44 example product", the data structure holds only one instance of the nut-bolt assembly, but three associations connecting it to the parent assembly.

A ramification of this BOM data structure is that the components of each assembly are indistinguishable within any assembly that includes it. Thus, although the L-bracket assembly contains three nut-bolt assemblies, each in turn containing a nut and a bolt, there is no way to distinguish any of the individual bolts or nuts within the L-bracket assembly except through their usage in the assembly.

E.1.3.2 Parts list data structures

The parts list data structure individualizes the relationship between lower-level parts of the product structure and higher-level assemblies in which they are contained.

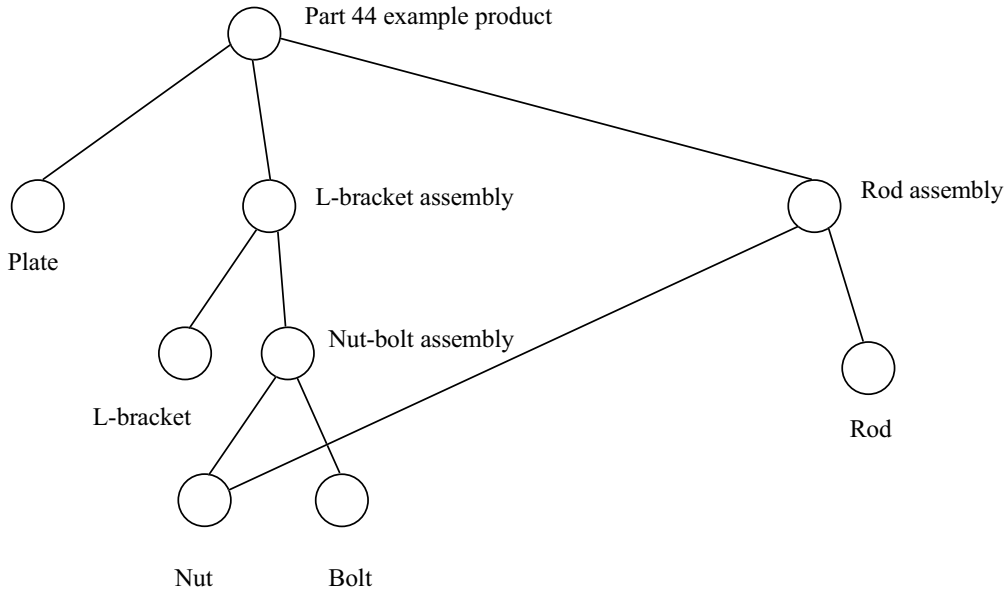

Figure E.3 - BOM data structure for part 44 example product

This data structure uses multiple occurrences of the same definition, or typical product, within the data structure at a single time. The representation of the usage of a particular definition is called an occurrence. An occurrence need not specify its definition counterpart. When there are two L-bracket assemblies within the "part 44 example product", the parts list data structure would include two different L-bracket occurrences which then permits the unique identification of the respective components of the L-bracket assemblies within the context of the overall "part 44 example product". Each L-bracket assembly occurrence is based on a typical L-bracket assembly definition that would contain all of the information that characterizes it. This parts list data structure can be seen in Figure E.4, where the L-bracket assemblies and the nut-bolt assemblies are individually represented. Each of these individually represented parts and assemblies are occurrences of a definition of those parts or assemblies. An explicit relationship to the definition could be specified.

E.1.3.3 Hybrid data structures

A hybrid data structure is a mixture of the BOM and parts list data structures. The mixture may have a definition in which the assembly structure lists all the occurrences that comprise its constituents. Conversely, an occurrence may have its constituents described as a single definition with multiple associations to describe each particular usage of the component definition within the assembly. Additionally, each occurrence that comprises an assembly may be decomposed implicitly by the decomposition of its associated definition.

E.1.4 Representation of product structure data structures using ISO 10303-44 entities

The BOM, parts list, and hybrid data structures are mapped into a combination of **product_definition** and **product_definition_relationship** (or subtypes) entities. For pure BOM structures, only the **quantified_-**

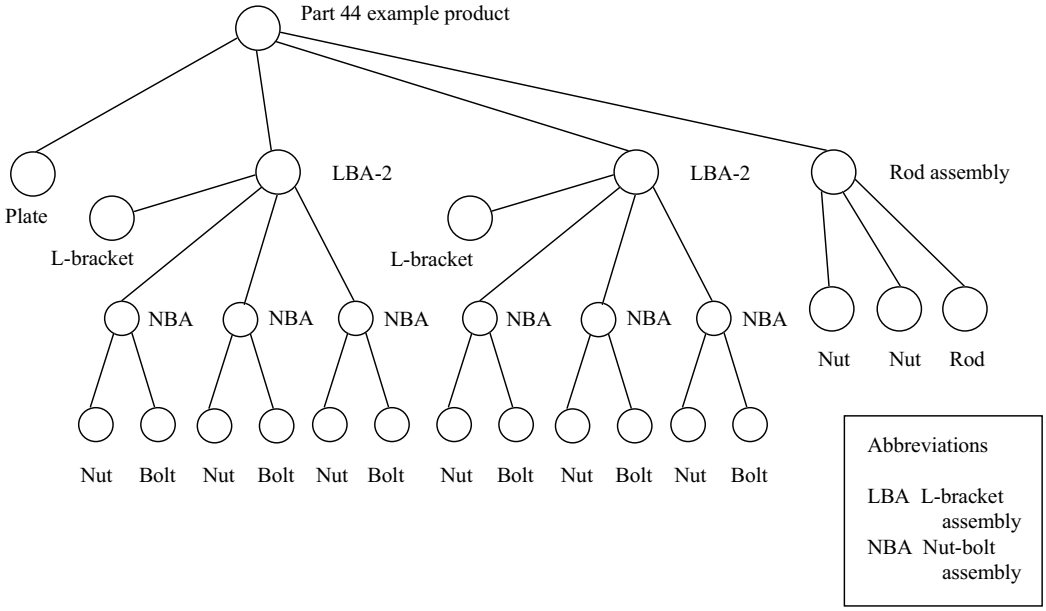

Figure E.4 - Parts list data structure for part 44 example product

assembly_component_usage, **next_assembly_usage_occurrence**, or **make_from_usage_option** subtypes of **product_definition_usage** need to be used, since the lower-level components are used only as "black boxes" to the parent assembly component. The "part 44 example product" does have two L-bracket assemblies, but from a BOM data structure point of view, both of these L-bracket assemblies are the same and it is not possible to distinguish their internal structure. For parts list structures, only the **specified_-higher_usage_occurrence** and **next_assembly_usage_occurrence** subtypes of **product_definition_usage** need to be used. In this case each of the L-bracket assemblies will be represented by unique instances of **product_definition**. For hybrid structures, all of the entities specified for BOM and parts list data structures may be used.

Figures E.5, E.7, and E.11 contain the symbology for the entities in this part of ISO 10303 which is used for the remainder of the figures in this annex. Rectangles represent the **product_definition** entities, and circles represent **product_definition_relationship** entities or its subtypes. The circles are labelled either with the first letter of the specific subtype or with "D-O" for a **product_definition_relationship** that specifies the relationship between an occurrence and its definition counterpart. In Figure E.5, there are instances of the **next_assembly_usage_occurrence** (label is "N"), the **quantified_assembly_component_-usage** (label is "Q"), and **make_from_usage_option** (label is "M") subtypes of **product_definition_usage**.

E.1.4.1 Representation with definitional BOM structures

For specifying a BOM, two methods can be used to show the quantity of a constituent within an assembly. The **quantified_assembly_component_usage** subtype of **product_definition_usage** can be used to specify the count of a component used in the assembly. Figure E.5 contains a subset of the BOM data structure. In the figure, there is a single relationship between the "part 44 example product" and the L-bracket assembly. This relationship is represented using a **quantified_assembly_component_usage** that specifies

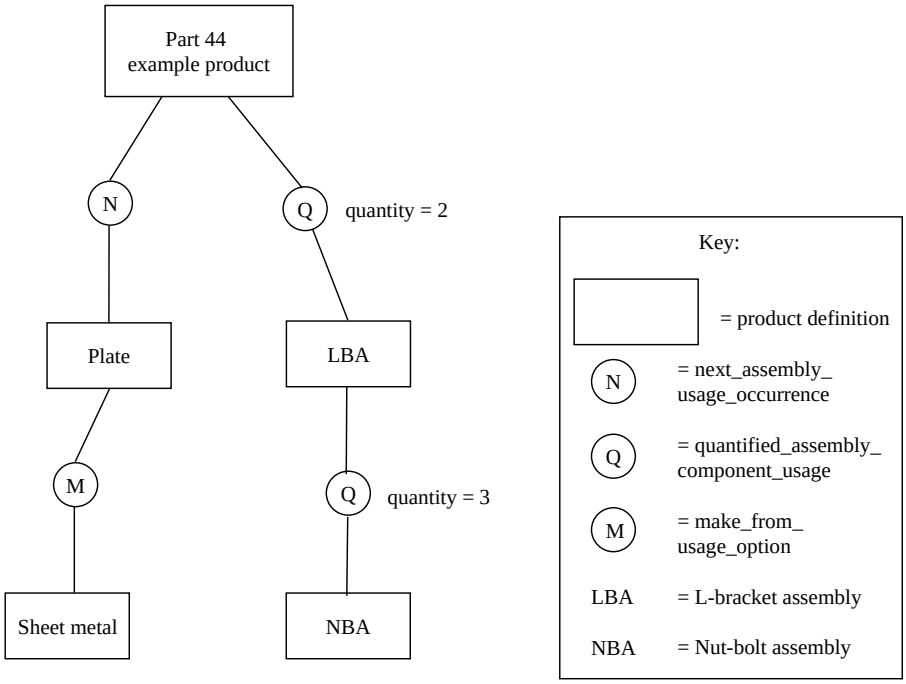

Figure E.5 - BOM data structure examples using ISO 10303-44 entities

a count of two for the L-bracket assembly. Similarly, there is a single relationship between the L-bracket assembly and the nut-bolt assembly. The **quantified_assembly_component_usage** specifies a count of three for the nut-bolt assembly.

The second method for specifying a BOM involves the identification of each relationship between the component and assembly. Figure E.6 presents the complete model for the "part 44 example product" in a BOM data structure using this part of ISO 10303. Note that individual **next_assembly_usage_-occurrence** entities represent each use of a component within a higher assembly. The quantity of components in each assembly can be calculated by counting the number **next_assembly_usage_-occurrence**s, rather than using a single **quantified_assembly_component_usage** entity that specifies the quantity.

E.1.4.2 Representation with definitional parts list structures

For specifying a parts list with definitional **product_definition** instances only, the **specified_higher_-usage_occurrence** subtype of **product_definition_usage** is used in order to make explicit the specific usage of a particular lower-level component within the overall context of a higher assembly. A parts list can specifically identify a sub-component of a lower assembly as distinguished from a similar sub-component of another lower assembly. This is possible because of the additional context information that

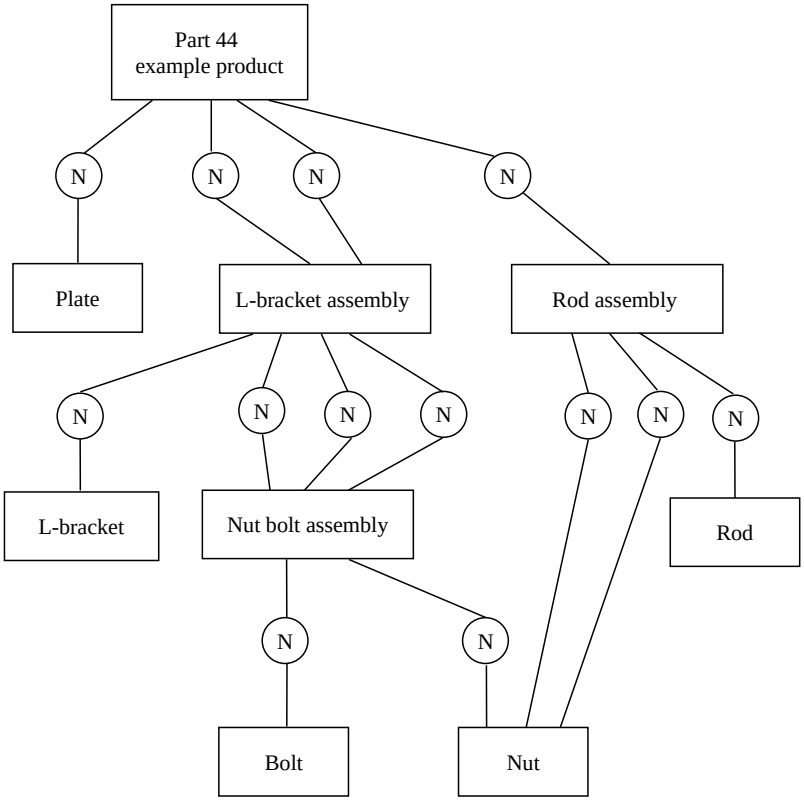

Figure E.6 - BOM data structure of part 44 example product

is provided by the **specified_higher_usage_occurrence** subtype of **product_definition_usage**. The additional context information is provided by specifying the specific immediate parent assembly and which specific top-level assembly. For instance, it is possible to identify the bolt on the third nut-bolt assembly of the second L-bracket assembly.

In Figure E.7, the additional information needed to define a parts list is represented by the additional subtype of **product_definition_usage**, namely the **specified_higher_usage_occurrence** entity, represented symbolically by the circle with label "S". Here, the **specified_higher_usage_occurrence** entity allows a reference to one specific nut-bolt assembly on only one of the two L-bracket assemblies. This is accomplished by having the **specified_higher_usage_occurrence** entity relate to only one of the two usages of the L-bracket assembly as represented by the **next_assembly_usage_occurrence** entity.

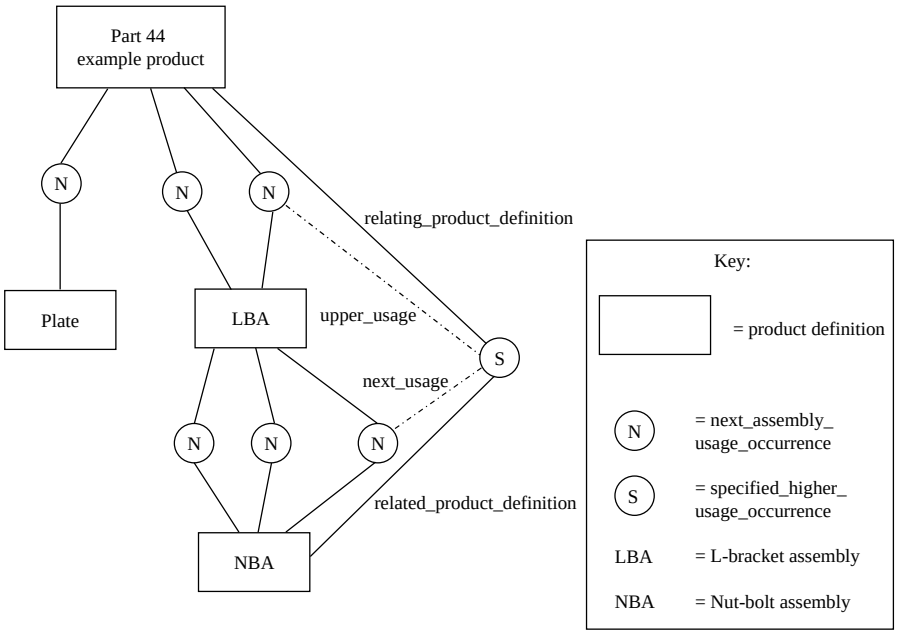

Figure E.7 - Parts list structure examples using ISO 10303-44 entities

Note that in this example, even though the parts list is implemented using the entities of this part of ISO 10303, it is done without having to create individual instances of the lower-level components. Rather, the additional identifying information is supplied through the existence of the intermediate **specified_higher_-usage_occurrence** and **next_assembly_usage_occurrence** entities.

In Figures E.8, E.9, and E.10, the representation of a number of lower-level components (rods, nuts, and assemblies) is presented within the context of higher-level assemblies as they would be represented using the entities of this part of ISO 10303. In each of these figures, the original BOM data structure is included within the large box; the additional entities needed to support the parts list aspects of the examples are placed outside the box, with their attributes referencing the original BOM.

In particular, Figure E.8 highlights the use of a **specified_higher_usage_occurrence** entity that represents how the Rod is used within the "part 44 example product" as specifically included in a Rod assembly entity. This is an example of a two-level component inclusion and as such, both the upper and next attributes of the **specified_higher_usage_occurrence** entity refer to **next_assembly_usage_occurrence** components in the BOM structure.

In Figure E.9, two different cases are represented. In one case, two **specified_higher_usage_occurrence** entities, located towards the top of the diagram, are included to represent the two nuts that are specifically used within the Rod assembly part of the "part 44 example product". Note that the extra relationship lines to the top-level "part 44 example product" definition and the low-level nut definition have not been included so as not to clutter up the diagram. In the other case, a **specified_higher_usage_occurrence** entity is used to represent one specific nut within an L-bracket assembly. Again, in both of these cases, only a two-level

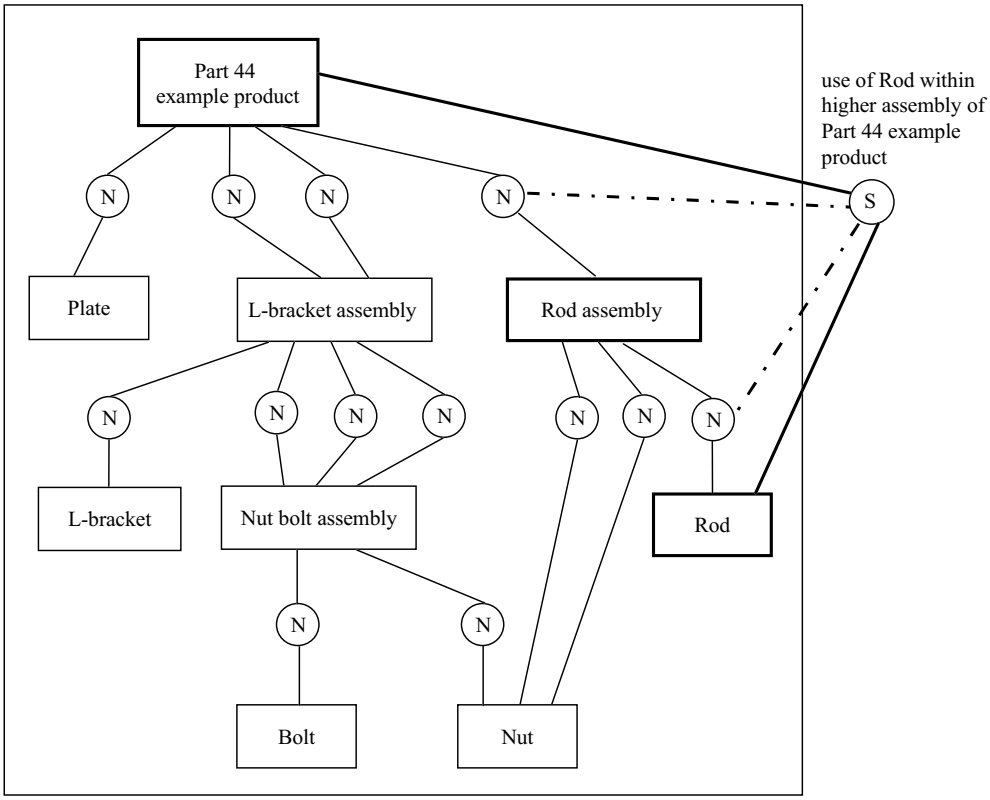

Figure E.8 - Simple specified higher usage occurrence

relationship between component and assembly is being represented and therefore the intermediate **product_definition_usage** entities are both **next_assembly_usage_occurrence**s.

Figure E.10 demonstrates two additional cases of parts list representation using a BOM data structure. First, three **specified_higher_usage_occurrence** entities are shown to relate the three different nut-bolt assemblies within the overall "part 44 example product". The second case shows a **specified_higher_-usage_occurrence** entity being used to represent one specific nut within a specific nut-bolt assembly within a specific L-bracket assembly of the overall "part 44 example product". This latter use of **specified_-higher_usage_occurrence** entity models a three-level structural relationship between one specific nut and the overall product. As such the associated **specified_higher_usage_occurrence** entity has its upper attribute referring to another **specified_higher_usage_occurrence** entity, namely, the entity that identifies the particular usage of the nut-bolt assembly within a particular L-bracket assembly. It is this level of support that allows the BOM model structure to support the parts list characteristic of individual occurrence labelling.

The increasing complexity of the parts list representation is a reflection of the complexity of the information within the data structure being managed. In fact, the increasing number of **specified_higher_usage_-occurrence** entities that would be included in a more complete model representation is directly proportional

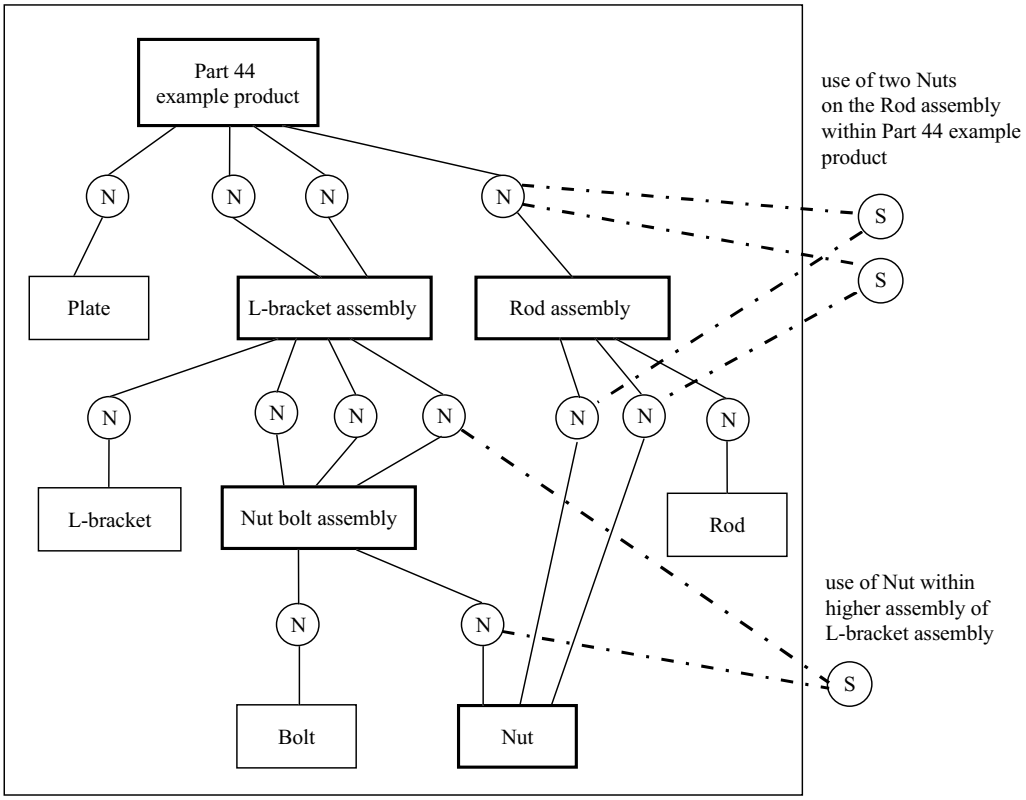

Figure E.9 - Specific usages with specified higher usage occurrence

to the number of possible paths that could be navigated through the parts list data structure model as seen in Figure E.7. What has been eliminated through the specification of a BOM data structure are the individual instances of each component, being replaced by individual **specified_higher_usage_occurrence** entities as required for a particular communication that provide multiple interconnections between a component and a higher-level assembly in which it is located.

E.1.4.3 Representation with occurrence parts list structures

A complete parts list representation of product structure may be specified using the entities in this part of ISO 10303. Figure E.4 depicts a representation of a data structure that captures each occurrence of a particular definition in the product structure. Each circle represents an occurrence of a typical part. Each occurrence is represented by an instance of a **product_definition** entity. Each **product_definition** would, in turn, have a relationship to the definitional **product_definition**. Each line represents an instance of **next_assembly_usage_occurrence**.

When the product structure is completely represented in this fashion, it may be traversed to reach any particular occurrence of a product in the structure. No **specified_higher_usage_occurrence** instances are

required because at any level, data may be associated directly with the individual instance of **product_-definition** that represents the particular occurrence of that product within the product structure.

E.1.4.4 Representation with hybrid structures

A hybrid structure may be created in which BOM and parts list representations are mixed to provide the capability to specify definitions and occurrences as required. Figure E.11 depicts a hybrid structure. Each definition and occurrence is specified by an instance of **product_definition**. In the figure, the definition of the "part 44 example product" consists of the definition of the plate and two occurrences of the L-bracket assembly. Each occurrence of the L-bracket assembly is related to the single BOM definition of the L-bracket assembly by an instance of **product_definition_relationship**, represented symbolically by the circle with label "D-O", that specifies it as a definitional occurrence. The definition of the L-bracket

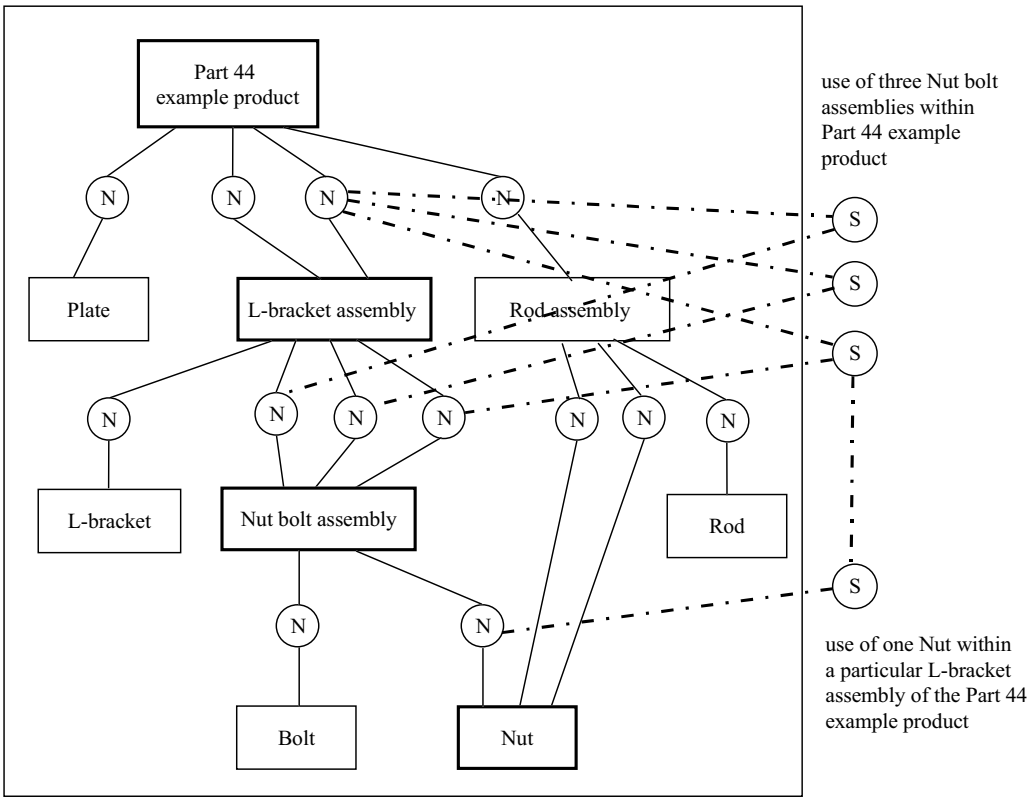

Figure E.10 - Complex specified higher usage occurrence

assembly is then specified using a parts list structure for the nut-bolt assembly in which three occurrences of that assembly are specified, and related to the L-bracket assembly using instances of **next_assembly_-**

usage_occurrence. Each occurrence of the nut-bolt assembly is then, in turn, related to the definition for that assembly of which they are each an occurrence, again with instances of **product_definition_-relationship** that specifies it as a definitional occurrence. Each assembly could be defined, if required, with a pure BOM representation using multiple instances of **next_assembly_usage_occurrence**.

A particular usage of the nut-bolt assembly within the context of the "part 44 example product" is not possible by simply specifying the particular occurrence **product_definition** within the hybrid structure. The **specified_higher_usage_occurrence** is necessary here as a result of the insertion of an intermediate definition in the product structure. To specify the usage of the third nut-bolt assembly within the second L-bracket assembly within the "part 44 example product", the **specified_higher_usage_occurrence** indicates the two **next_assembly_usage_occurrence**s that specify the use of the component in the assembly at each intermediate level. Since the definitions and occurrences are not directly related to each other in the hybrid case, the **specified_higher_usage_occurrence** specifies the usage of each occurrence within the product structure of the definition.

E.1.5 Generating product structure reports

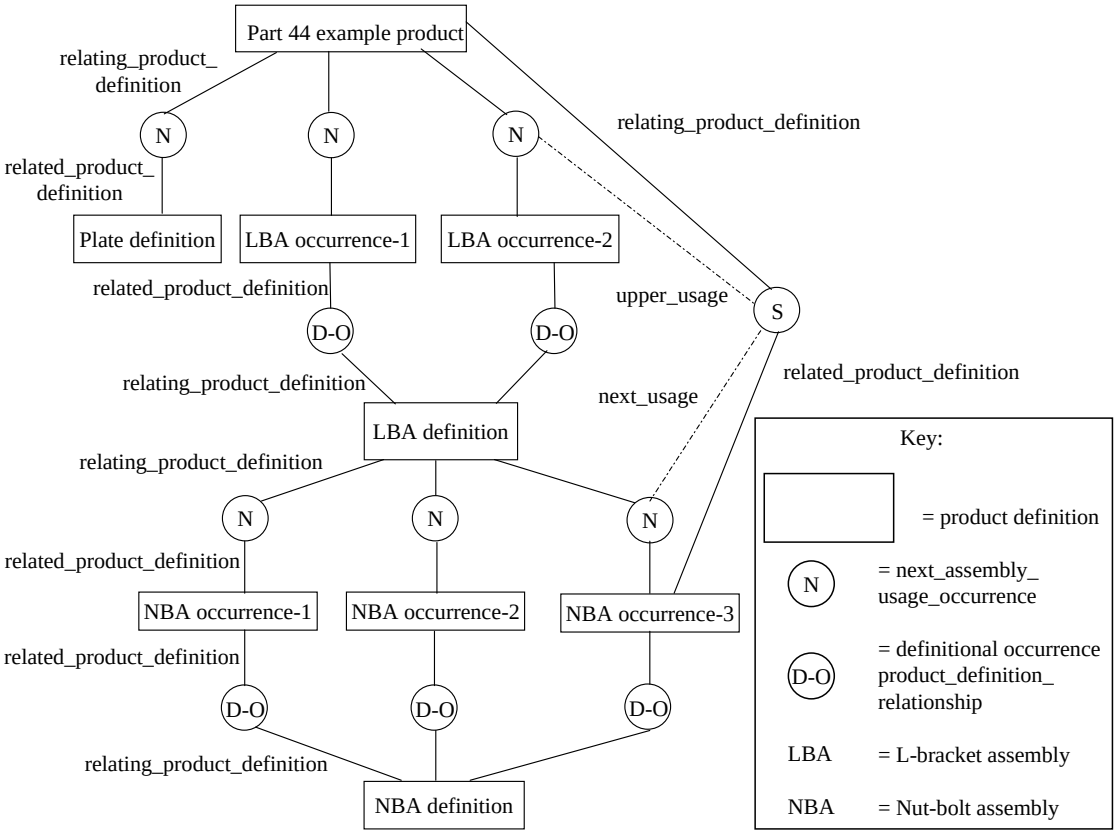

Figure E.11 - Hybrid structure of part 44 example product

Once the actual product structure has been represented using the entities of this part of ISO 10303, it is possible to generate the various product structure reports mentioned earlier in this annex. The algorithm consists in identifying the top-level product about which the report is to be generated, then iterating through its constituent sub-assemblies via the **quantified_assembly_component_usage**, **next_assembly_usage_occurrence**, or **make_from_usage_option** entities, invoking indentation, numeric values, and labelling where appropriate, based on the information available in the entities as they are traversed. Well known tree-traversal algorithms exist for descending through several levels of the structure to provide the necessary multi-level reports. In addition, as the traversal is taking place, the additional structural information available through the **specified_higher_usage_occurrence** entities can be accessed and applied to the resulting report as necessary.

Report generation thus becomes an exercise in traversing between **product_definition** entities via various subtypes of the **product_definition_usage** entity type, i.e., **make_from_usage_option**, **quantified_assembly_component_usage**, **next_assembly_usage_occurrence**, and **specified_higher_usage_occurrence**, utilizing the added information provided uniquely by each such entity type.

E.1.6 Product structure summary

In summary, this annex has presented a number of representations of the "part 44 example product" that support a BOM data structure, a parts list data structure and a hybrid data structure. From the appropriate use of each of these representations, it is possible to generate the various product structure reports provided in the earlier section of this annex. As seen from the examples, one of the primary features of the entities defined in this part of ISO 10303 is that it permits the data modeler to model various types of product structure data structures (e.g., BOM, parts list, etc.) using the same primitive entities, thus facilitating meaningful information exchange between users without resorting to costly data model mapping tasks. As an example of this specific feature, it was demonstrated that the entities of this part of ISO 10303 support an incremental evolution from a pure BOM data structure implementation to a parts list data structure implementation by adding only instances of the **specified_higher_usage_occurrence** entity without altering any of the existing BOM data structure implementation.

E.2 Product features and product concepts

A product concept may allow several valid product variations based on dependencies and conditions specified for the different features that are identified for it. In this annex, examples of different types of dependencies between product features are provided with the use of this part of ISO 10303 to represent these dependencies.

A particular model manufactured by a car company is represented by a **product_concept**. The model is manufactured with a diesel engine (DE) or a spark ignited engine (SI). Each of these is represented by instances of **product_concept_feature**. The global condition "DE XOR SI" can be formulated to ensure that in each manufactured car there is exactly one of these engines. This condition is represented by a **conditional_concept_feature** referencing a **concept_feature_relationship_with_condition** with the conditional_operator being a **concept_feature_operator** with name "XOR". The **concept_feature_relationship_with_condition** references a **product_concept_feature** with id "DE" as relating_product_concept_feature and a **product_concept_feature** with id "SI" as related_product_concept_feature.

Features of a particular model of car can include an air conditioning system (AC) and electric windows (EW). If both of these features are ordered, however, a stronger battery (SB) is also required. Each of these features of the car is represented by instances of **product_concept_feature**. The condition that a stronger battery is required when air conditioning and electric windows are ordered is represented by instances of **conditional_concept_feature**. Different logical formations using **concept_feature_operator** can be defined to represent this condition.

Figure E.12 shows one method of representing this condition with the use of the binary operators AND, OR, and NOT. The boolean expression would be "(((AC AND EW) AND SB) OR (NOT (AC AND EW)))". A **concept_feature_relationship_with_condition**, with a **concept_feature_operator** having the name "AND", establishes the relationship that both the air conditioning system and electric windows are

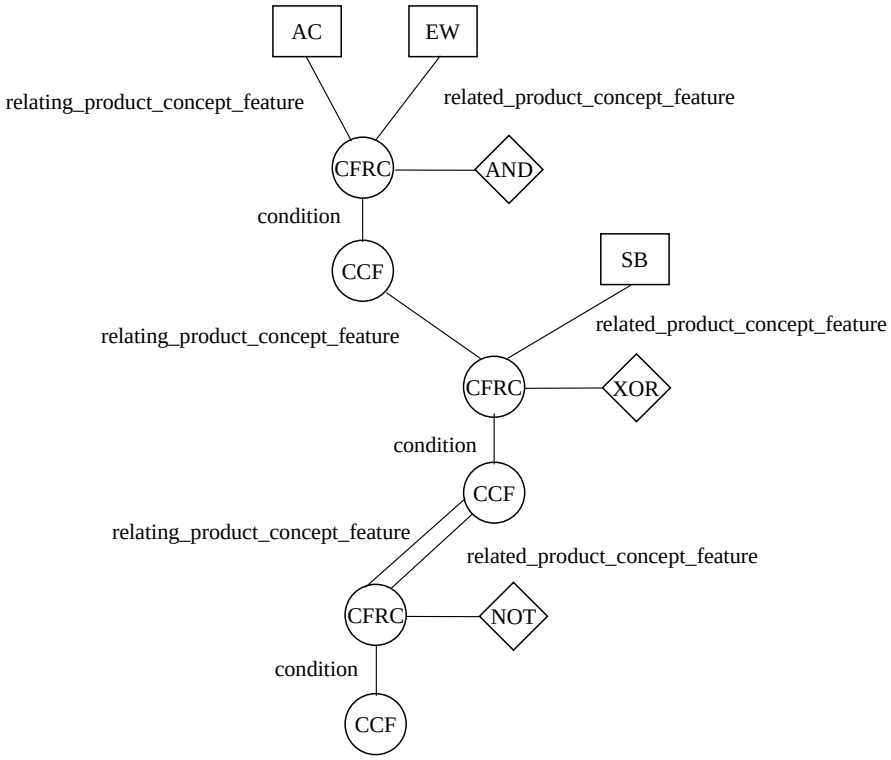

Figure E.12 - Alternate binary operators for conditional_concept_feature

requested. This relationship is the condition for a **conditional_concept_feature**. The **conditional_-concept_feature** is related to the stronger battery with an AND operator to say that all three features are present. The **conditional_concept_feature** is also used in a relationship with itself, by convention, to establish the condition of negation. The resulting **conditional_concept_feature**s are related with an OR operator to say either all three features are present or both the air conditioning system and electric windows are not.

Figure E.13 shows a method of representing a similar condition where the additional constraint that that stronger battery is only included in the car if both the air conditioning system and electric windows are present. This method uses the binary operators AND, XOR, and NOT. The boolean expression would be "NOT ((AC AND EW) XOR SB)". A **concept_feature_relationship_with_condition**, with a **concept_-feature_operator** having the name "AND", establishes the relationship that both the air conditioning system and electric windows are requested. This relationship is the condition for a **conditional_concept_-feature** that is related to the stronger battery with an XOR operator to say that either the air conditioning system and electric windows are present or the stronger battery is present. The resulting **conditional_-concept_feature** is related with itself, by convention, to establish the condition of negation to establish the required combination of features.

Figure E.14 shows a method of representing the same condition using the binary operator AND and by introducing an operator for implication. The expression would be "(AC AND EW) IMPLIES SB". A

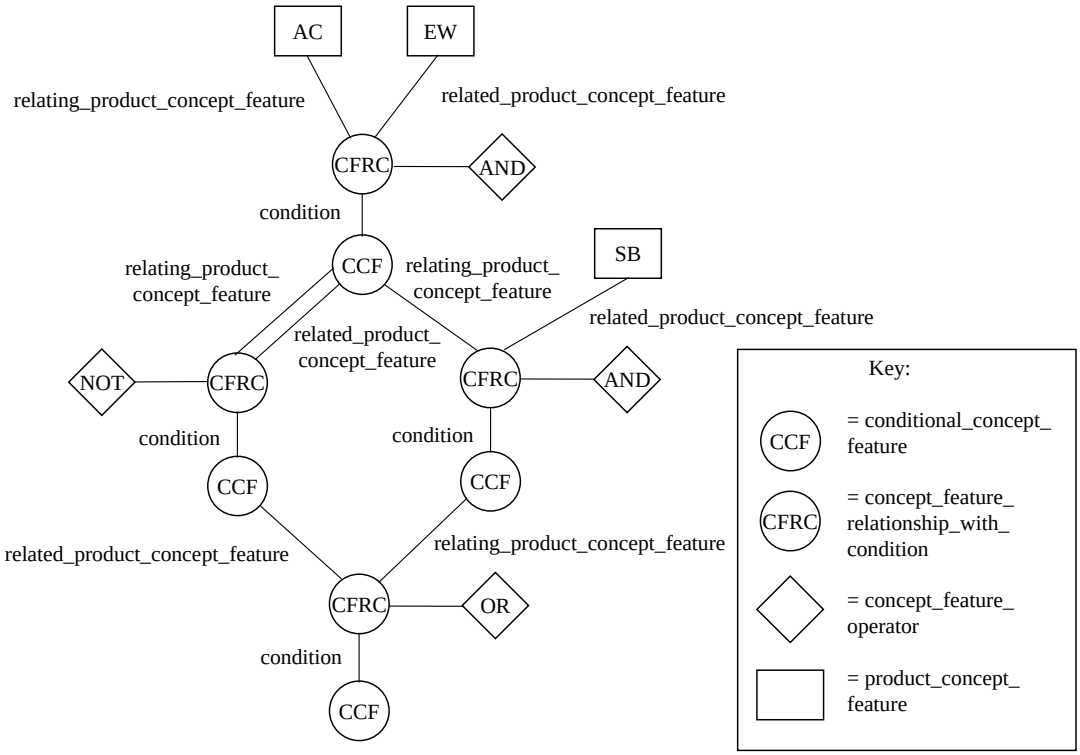

Figure E.13 - Binary operators for conditional_concept_feature

concept_feature_relationship_with_condition, with a **concept_feature_operator** having the name "AND", establishes the relationship that both the air conditioning system and electric windows are requested. This relationship is the condition for a **conditional_concept_feature** that is related to the

stronger battery with the newly introduced IMPLIES operator. This relationship establishes the condition that either all three features are present or both the air conditioning system and electric windows are not.

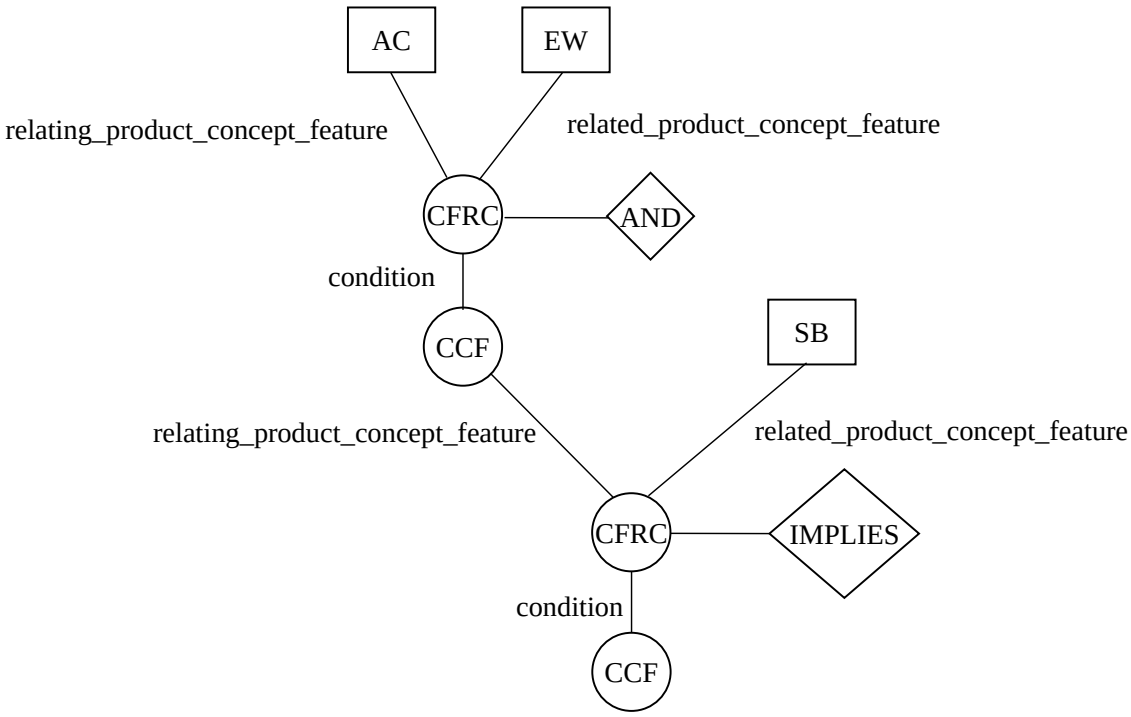

Figure E.14 - Defined operators for conditional_concept_feature

Bibliography

[1] Hartsfield, N and Ringel, G.; Pearls in Graph Theory: A Comprehensive Introduction. Academic Press, Inc. San Diego, 1990.

[2] ISO 10303-42:1994, *Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resources: Geometric and topological representation.*

[3] ISO/DIS 10303-43:—[1)], *Industrial automation systems and integration — Product data representation and exchange — Part 43: Integrated generic resource: Representation structures.*

[1)] To be published.

Index

